

Optimizasyon Teknikleri

Ders Notları

Doç. Dr.
İBRAHİM KÜÇÜKKOÇ
<http://ikucukkoc.baun.edu.tr>

Balıkesir Üniversitesi
Endüstri Mühendisliği Bölümü

EMM3208

Optimizasyon Teknikleri Ders Notları

Doç. Dr. İbrahim Küçükkoç
<http://ikucukkoc.baun.edu.tr>
ikucukkoc@balikesir.edu.tr

12.03.2020

Ders Planı, Değerlendirme Kriterleri, Yararlanılacak Kaynaklar

1

DERSLE İLGİLİ BİLGİLER

EMM3208 Optimizasyon Teknikleri

Dersin Amacı:

- Optimal karar vermeyi sağlayan matematiksel modellerin çözümünde kullanılacak yöntem ve teknikleri sunmak ve gerçek hayatta karşılaşılan uygulama alanlarını tanıtmaktır.

Dersin Web Sayfası: <http://ikucukkoc.baun.edu.tr/lectures/EMM3208>

- Ders notları düzenlendikçe bu sayfadan yayınlanacaktır.

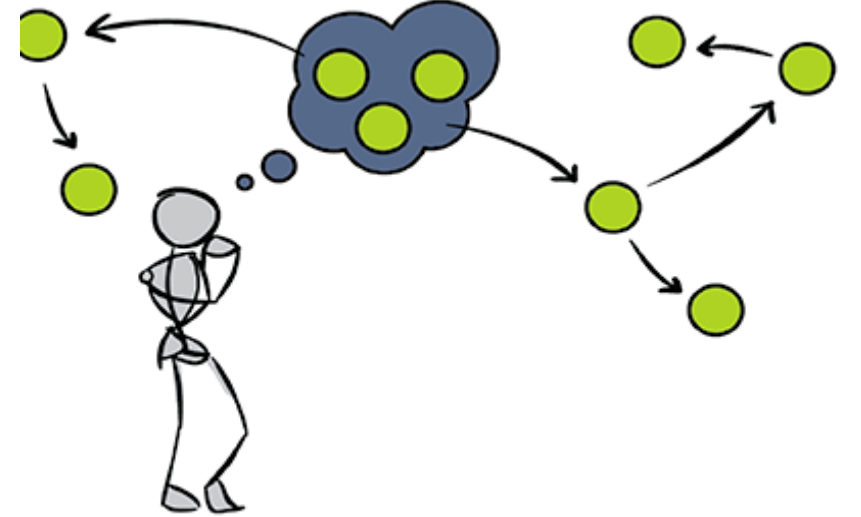
Değerlendirme:

- Vize (%40) + Final (%60)
- Final sınav puanının 25 puanlık kısmı dönem içerisinde yapılacak olan projeden alınacaktır. Projeler maksimum 2 kişilik gruplar halinde yapılabilir.

DERSE İLGİLİ BİLGİLER

İÇERİK

- 1 Optimizasyona Giriş, Temel Kavramlar
- 2 Optimizasyonda Kullanılan Yöntemler (Simplex, Grafik Yöntem vs.), Excel Solver
- 3 GAMS - Kullanıcı Arayüzü ve Özellikleri, Program Yapısı
- 4 GAMS - Çözücülerinin Tanıtımı ve Basit Doğrusal Modellerin Çözümü, Örnekler
- 5 GAMS - Set kullanımı, Veri Girişi (Listeler, Tablolar, Doğrudan Atama), Örnekler
- 6 GAMS - Değişkenler ve Eşitlikler, Örnekler
- 7 GAMS - Amaç Fonksiyonu, Örnekler
- 8 GAMS - *Model* ve *Solve* Belirteçleri, Örnekler
- 9 GAMS - *Display* Belirteçleri, Örnekler
- 10 GAMS - 'lo, .l, .up ve .m' Veritabanı, Örnekler
- 11 GAMS - Çıktı Raporunun İncelenmesi, Hatalar, İstatistikler
- 12 GAMS - Şartlı İfadeler
- 13 GAMS - Proje Çalışması ve Sunumu
- 14 GAMS - Proje Çalışması ve Sunumu



DERSLER İLGİLİ BİLGİLER

Derse Katılım:

- Derslere zamanında gelmeniz gerekmektedir.
- 5 hafta ve üzeri devamsızlık yapan öğrenciler devamsızlıktan bırakılacak ve final sınavına alınmayacaktır.
- Derste cep telefonu vb. konuyla alakasız materyallerle ilgilenilmemesi beklenmektedir

Gerekli Programlar:

- General Algebraic Modelling System (GAMS): <https://www.gams.com>

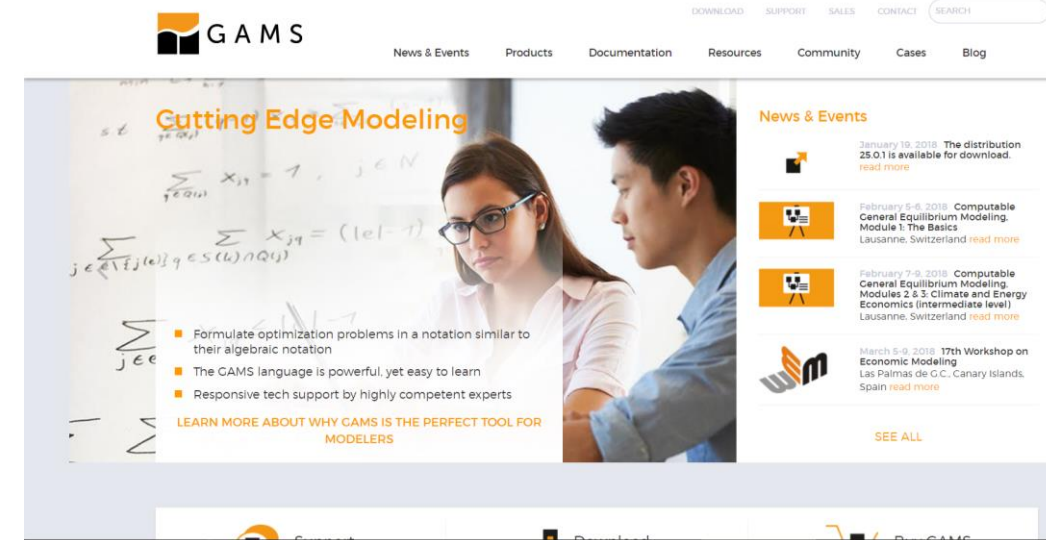
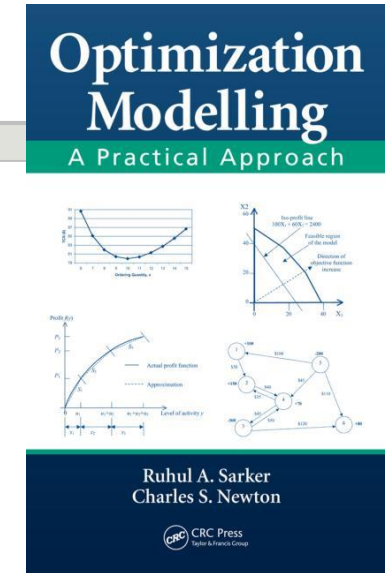
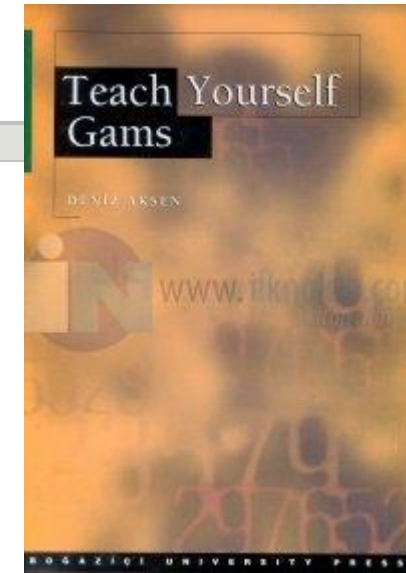
GAMS ve Çözücüler Hakkında Dokümantasyon:

- https://www.gams.com/latest/docs/UG_MAIN.html
- https://www.gams.com/latest/docs/UG_Tutorial.html
- https://www.gams.com/latest/docs/S_MAIN.html

DERSLER İLGİLİ BİLGİLER

Yararlanılacak Kaynaklar:

- Teach Yourself GAMS, Deniz Aksen, Boğaziçi Üniversitesi Yayınları, 1998
- [Optimization Modelling: A Practical Approach](#) Ruhul A. Sarker, Charles S. Newton, CRC Press Book, 2008
- [GAMS Web Sayfası ve Dokümantasyonu](#)
- [GAMS A User's Guide, Tutorial by Richard E. Rosenthal USA, 2007](#)
- [GAMS Basics, Stefan Vigerske](#)
- [Introduction to GAMS: Summation Notation with GAMS](#) Dhazn Gillig, Bruce A. McCarl, Texas A&M University



Optimizasyona Giriş

Temel Kavramlar

Excel Solver Kurulumu ve Kullanımı

Örnekler

Optimizasyona Giriş & Temel Kavramlar



Optimizasyon Nedir?

Optimizasyon, en genel anlamıyla, bir sistemde, belirli kısıtlar altında, belirlenmiş bir amaç fonksiyonunun değerinin en iyilenmesi amacıyla karar değişkenlerinin alacağı değerleri belirleme işlemidir.

Diğer bir ifade ile, istenen bir çıktıyı elde etmek amacıyla, sistem girdilerinin ve/veya bu girdilerin değerlerinin ne olacağını belirleme sürecidir.

Amaç fonksiyonunun en iyilenmesi demek, problemin türüne göre en küçüklenmesi (minimizasyon) veya en büyüklenmesi (maksimizasyon) olabilir.

Örneğin bir çizelgeleme probleminin, geciken iş sayısını en küçükleme amacıyla çözülmesi bir optimizasyon problemidir. Aynı şekilde, bir montaj hattı dengeleme probleminin istasyon sayısını minimize etmek (veya hat etkinliğini maksimize etmek) amacıyla, veya bir ekonomik stok miktarı belirleme probleminin toplam maliyeti minimize etmek amacıyla çözülmesi birer optimizasyon işlemidir.

Optimizasyona Giriş & Temel Kavramlar



Simülasyon, girdilerin bilindiği bir sistemde, çıktının tahmin edilmesi, belirlenmesi, sürecidir. Örneğin, siparişlerin geliş zamanının ve sistemde geçireceği sürelerin bilindiği bir üretim sürecinin simülasyonu sonucu, üretimin ne zaman tamamlanacağı, sistemdeki boş zamanlar, sistemin etkinliği vs. büyük oranda hesaplanabilir (tahmin edilebilir).

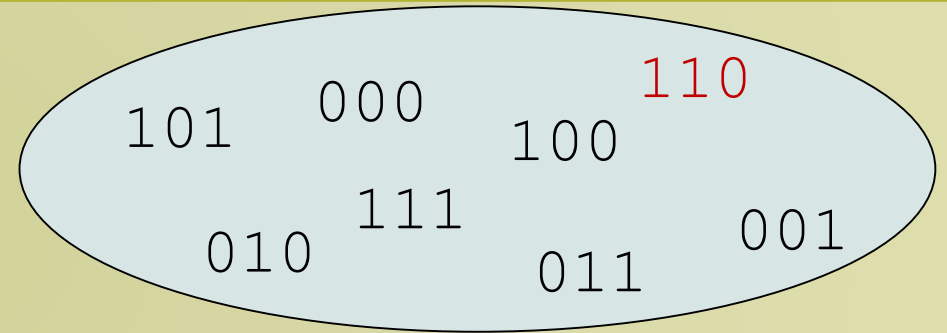
Optimizasyon ise, istenen çıktıyı elde edebilmek amacıyla, girdilerin veya bu girdilerin değerlerinin ne olacağının belirlenmesi sürecidir. Örneğin, istenen yüzey pürüzlülüğünün elde edilebilmesi amacıyla, bir 3B yazıcının parametrelerinin ne olacağının belirlenmesi optimizasyona örnektir.

Optimizasyona Giriş & Temel Kavramlar

Karar değişkeni, problem çözüldüğü zaman değeri belirlenecek olan sistem ögesidir. Yani aslında karar değişkenlerinin alacağı değerlerin belirlenme süreci, problem çözme sürecidir. Karar değişkenlerinin alacağı değerler, amaç fonksiyonunun değerinin ne olacağına etki eder. Fakat bu süreçte bazı kısıtların sağlanması gerekmektedir.

Çözüm uzayındaki, problemin kısıtlarını sağlayan her hangi bir çözüm, **uygun çözüm** olarak adlandırılır. **Optimum çözüm** ise, uygun çözümler arasından, belirlenen amaca göre en iyi çözümü ifade etmektedir.

20kg, 75kg ve 60kg ağırlıklarında üç adet nesne var. 100kg'a en yakın olacak şekilde hangi nesneleri almaliyiz?



Amac Fonksiyonu: |toplam ağırlık - 100|

Optimizasyona Giriş & Temel Kavramlar

Optimizasyon işleminde kullanılan teknikler:

Analitik Yöntemler (matematiksel modelleme, dal-sınır algoritması, vb.):

Toleranslar dahilinde optimum çözümü verirler. Büyük boyutlu problemlerde sonuca ulaşmak çok uzun sürer veya ulaşılamaz.

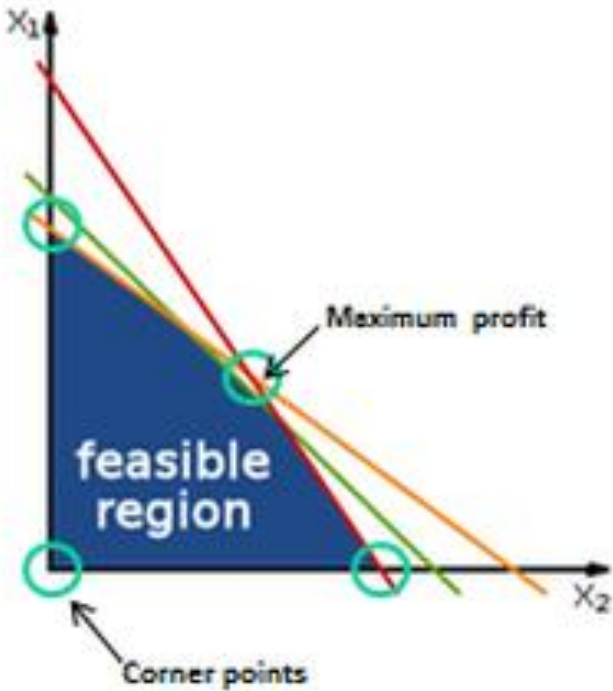
Sezgisel Yöntemler:

Probleme özgü çözüm yöntemleridir ve belirli bir algoritmayı takip ederler. Optimum çözümü garanti etmemekle birlikte analitik yöntemlere göre daha hızlı çözüm üretirler. Örn. akış tipi çizelgeleme problemleri için Johnson algoritması, montaj hattı dengeleme problemleri için pozisyon ağırlığı yöntemi, vb.

Meta-sezgisel Yöntemler:

Belirli algoritmaların, çözülecek problem yapısına uyarlanması ile elde edilen çözüm yöntemleridir. Örn. genetik algoritma, karınca koloni algoritması, tavlama benzetimi algoritması, vb.

Doğrusal Programlama



Bir doğrusal programlama problemi genel itibari ile amaç fonksiyonu ve doğrusal sınır/sınırların yer aldığı iki kısımlı bir matematiksel ifadedir. Bu matematiksel ifade ile bir amaç **maksimize** ya da **minimize** edilir.

Doğrusallık ifadesi modelde yer alan tüm değişkenler (fonksiyonlar) arasındaki ilişkinin doğrusal olmasından kaynaklanmaktadır.

Doğrusal sınırların oluşturduğu kesişim kümesinden yola çıkılarak mümkün çözümler ya da uygun çözüm alanı belirlenir. Belirlenen uygun çözüm alanı ise amaç doğrultusunda eniyilenmeye çalışılır.

Doğrusal Programlama

- Doğrusal programlama (DP) ile bağımsız değişkenlerden oluşan bir dizi fonksiyon ile yine bir dizi bağımsız değişkenlerin bir fonksiyonu olan bağımlı değişkenin optimal değeri belirlenmeye çalışılır.
- Bir başka ifade ile doğrusal programlama, belirli bir amacı eniyilemek maksadıyla sınırlı kaynakların nasıl dağıtılması gerektiğine çözüm arayan bir karar verme aracıdır.

Doğrusal Programlama Modelinin Kurulması

$$\text{Max } Z = \sum_{j=1}^n c_j x_j$$

$$\sum_{j=1}^n a_{ij} x_j (\leq, =, \geq) x_j \quad i = 1, 2, \dots, m$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n$$

Burada;

Z : optimize edilmeye çalışılan amaç fonksiyonu

x_j : j . karar değişkenine atanacak değer yada belirlenecek değişken

c_j : 1 br j . karar değişkeninin amaç fonksiyonuna katkısı

a_{ij} : Sınır matrisi A 'yı oluşturan teknoloji katsayıları

b_i : i . kaynak için gerekli olan miktar (sağ taraf sabiti)

DP Modellerinin Dayandığı Varsayımlar

Doğrusallık:

DP modeli değişkenler arasındaki ilişkilerin doğrusal olduğu varsayımına dayanır. Bu varsayım, bütün ilişkilerin birinci dereceden fonksiyonlar ile ifade edilmesinden kaynaklanır. Her bir karar değişkeninin amaç fonksiyonuna katkısı, diğer karar değişkenlerinin değerlerinden farklı olarak o karar değişkeninin değeri ile orantılı olması, doğrusallığı göstermektedir. Gerçek dünyada bu duruma uygun düşen örnekler vardır. Mesela, bir makinenin kapasitesi saatte 300 birim ise, bu makineden 2 saatte 600 birim, 3 saatte ise 900 birim ürünün elde edilmesi doğrusallık kavramını açıklamaktadır.

Doğrusallık varsayımı aslında temel olarak orantılı olma ve katkı olmak üzere iki özellik ile ilgilidir:

- Orantılı olma: Bu özellik, her bir karar değişkeninin gerek amaç fonksiyonuna gerekse de tüm kısıtlara etkisinin, söz konusu değişkenin değeriyle doğru orantılı olması gerektiğini ifade eder.
- Katkı: Bu özellik de, kısıtlardaki ve amaç fonksiyonundaki tüm değişkenlerin toplam katkısının bu değişkenlerin tekil (bireysel) katkılarının toplamından oluşması şartını koşar.

DP Modellerinin Dayandığı Varsayımlar

Toplanabilirlik:

Toplanabilirlik varsayımı doğrusallık varsayımının doğal bir sonucudur. Kıt kaynakların kullanılması çerçevesinde toplanabilirlik varsayımı düşünüldüğünde, rakip faaliyetler tarafından birlikte kullanılan toplam kaynak miktarının bu rakip faaliyetlerin teker teker kullandıkları miktarların toplamına eşitlenmesidir. Bu varsayım amaç fonksiyonu çerçevesinde düşünüldüğünde ise, bu fonksiyonun bağımlı değişkeninin değeri tek tek faaliyetlerden kaynaklanan kar katkılarının toplamına eşitlenmesidir.

Bölünebilirlik:

DP modelinin dayandığı bu varsayıma göre her bir rakip faaliyetin sonsuz derecede bölünebilir olmasıdır. Bu varsayım çerçevesinde, karar değişkenleri tamsayılı değerlerin yanında kesirli değerlerde alabilmekte, kıt kaynaklar kesirli miktarlarda kullanılabilir. Kesirli sonuç istenmeyen durumlarda da DP modeli kullanılabilir fakat elde edilen sonuç yuvarlanır. Bunun dışında karar değişkenlerinin zorunlu olarak tamsayılı değerler almaları istenilirse tamsayılı doğrusal programlama kullanılır.

DP Modellerinin Dayandığı Varsayımlar

Belirlilik (Kesinlik):

DP modelinde rakip faaliyetlerin amaç fonksiyonuna katkılarının, kullandıkları kaynak miktarlarının ve kısıtlı kaynakların mevcut miktarlarının önceden kesinlikle bilindikleri varsayılmaktadır. Gerçek hayattaki problemlerin aslında küçük bir bölümü bu varsayımı sağlamaktadır. Doğrusal programlama genellikle gelecekteki faaliyetlerin seçiminde kullanılır. Bu yüzden de parametre değerleri gelecekteki koşullar dikkate alınarak belirlenir. Bundan dolayı, durumun bu şekilde olması halinde kullanılan çözüm yöntemi ise optimal çözüm bulunduğundan sonra bu çözümün parametrelere olan duyarlılığını göstermede kullanılacak olan duyarlılık analizidir.

Görüldüğü gibi bir DP probleminde bu varsayımların tamamının sağlanması halinde modelin kurulması söz konusu olacaktır. Bu varsayımlar dikkate alındığında her karar problemine DP modelinin uygulanamayacağı açıkça görülmektedir.

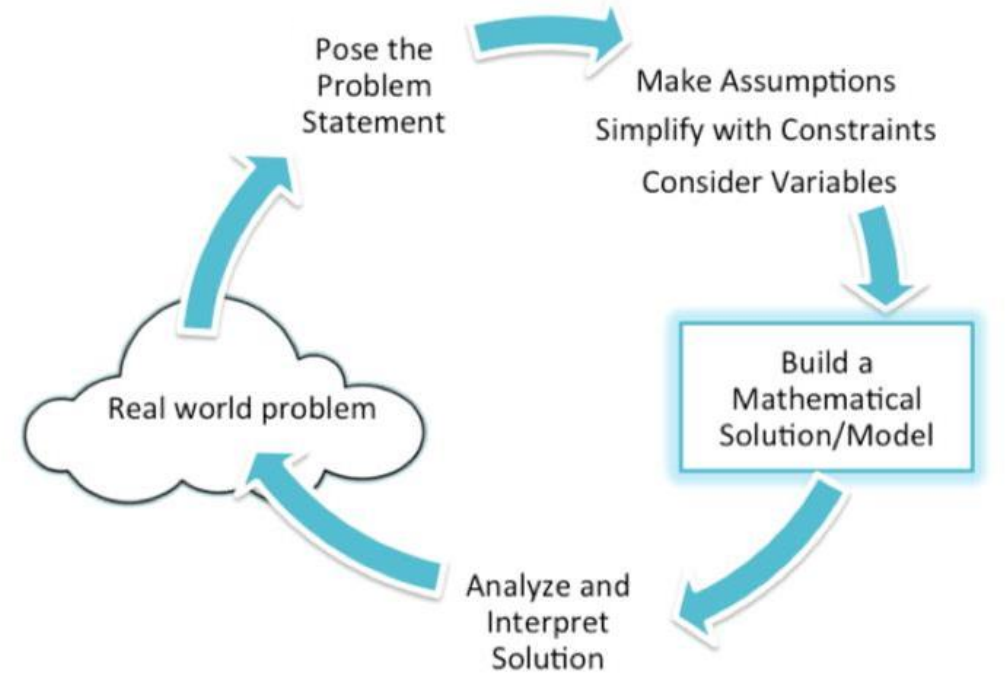
DP'nin Sanayi Uygulamaları

Doğrusal programlama birçok mümkün çözümler arasından en uygun olanı seçmede kullanılan bir grup matematik teknik olarak tanımlanır. Uygulamada karşılaşılan problemlerin bir çoğu bu tip olduğundan doğrusal programlama, işletmelerde değişik alanlara başarıyla uygulanmıştır. Buna ilave olarak ekonomi tarım ve mühendislik gibi değişik bilimlerde yapılmış çok sayıda uygulama çalışmaları vardır.

Bilgisayarların modellerin çözümünde kullanılması uygulama çalışmalarını hızlandırmıştır. Bu yüzden doğrusal programlama hem planlama hem de günlük programlama meselelerinde başarıyla kullanılmıştır ve gelecekteki uygulama sahasının daha da genişleyeceğine şüphe yoktur. Geçen bir kaç yılda bu uygulamalar hızla geliştirilmiş ve sayıları artmıştır. Doğrusal programlama uygulamasında önem endüstri uygulamalarına geçmiştir. Çünkü doğrusal programlama işletmelerde yöneticilere değişik biçimlerde yararlı ve yardımcı olmaktadır. Öncelikle DP işletmedeki üretim ve işlemler hakkında önceden bilgi vermektedir. İkincisi, endüstriyel sistemlerin yapısını matematik olarak araştırmaya, modeller kurmaya zorlamaktadır. Üçüncüsü de işletmede verimliliğin artması için yöneticilerin elinde önemli bir araç olmaktadır.

DP'nın Sanayi Uygulamaları

Doğrusal programlama tekniğiyle endüstride karşılaşılan birçok problem kolaylıkla çözülebilmektedir. Ancak dikkat edilmesi gereken nokta, bu problemlerin matematiksel ifadesinde, doğrusal programlamanın gerçek sorunu tam olarak yansıtmayabileceğidir. Çoğu kez modeller bazı varsayımlar altında yapılan bir yaklaşımdır. Bununla beraber modeller işletme ve yönetim için oldukça yararlı sonuçlar vermektedir. Aslında mühendislik gibi uygulamalı bilimlerde de gerçek sorunu kesin bir şekilde ifade eden modeller oldukça azdır. Önemli olan modelin tutarlı bilgiler temin etmeye yeterli doğrulukta olmasıdır.



DP – Karışım Problemleri

Doğrusal programlama modellerinin uygulamaları göstermiştir ki temelde bir kaç ana model vardır. Diğer uygulamalar bu ana modellerin kombinasyonu ya da değişik konularda uygulamalarıdır.

Uygulamaları değişik şekillerde gruplandırma yapılmaktaysa da temel olarak endüstridekiler üç gruba ayrılabilir:

- a) Değişik üretim konularında üretim ve stok planlama uygulamaları.
- b) Petrol ve diğer sanayi sektörlerinde optimum karışımın tayinindeki uygulamalar.
- c) Doğrusal programlama modelinin özel bir hali olan ulaştırma modelinin uygulamaları.

Örnek olarak karışım problemleri üzerinde duralım.

DP – Karışım Problemleri

Genel olarak karışım problemlerinde bir ya da daha fazla ürünü elde etmek için çok sayıda ara ürün karıştırılır. Çoğu kez hammaddeler açısından ürünün kalitesini ve miktarını etkileyen sınırlayıcı şartlar vardır. Buna ilaveten son ürünün elde edilmesinde değişik şartlara bağlı hammaddeler çok sayıda değişik biçimde ve miktarda karıştırılabilirler. Bu yüzden karışım probleminin verilen bir amaç fonksiyonunu optimize edecek bir biçimde düzenlenmesi gerekir. Bu genellikle belirli sınır ve şartları sağlayan bir maliyet minimizasyonudur. Karışım problemlerine örnek yalnızca petrol endüstrisi değildir. Yem, gıda, demir-çelik metalürji gibi sanayi kollarında da karışım problemleriyle karşılaşılır.

DP – Karışım Problemleri

Doğrusal programlama değişik oranlarda doğrusal değişen karışımların, karışım işlemleri için uygun bir yapı gösterir. Bu, hiçbir zaman karışımın bütün fiziki miktarlarının doğrusal değiştiği anlamına gelmez. Bu yüzden doğrusal programlama doğrusal olmayan durumlarda uygulandığında yalnızca yaklaşık bir sonuç verir. Birçok durumda doğrusallık kabulü karışım işlemlerinin incelenmesinde, çok yararlı ve yeterli sonuçlar almayı sağlar.

Doğrusal programlamanın ilk tutarlı ve yararlı uygulaması petrol rafinerilerinin programlamasında olmuştur.

Yakın zamanlarda birçok petrol şirketi bu konuda geniş çalışmalar yapmıştır.

DP – Karışım Problemleri

Rafineri işlemleri ham petrolün havada damıtılmasıyla başlar. Bu damıtma sonucu çok sayıda ara ürün ortaya çıkar. Diğer ürünler aynı hammadde ya da ara ürünlerin değişik oranlarda karıştırılmasından elde edilir. Bu yüzden rafineri işlemlerinin optimizasyonu oldukça karışıktır. Charnes, Cooper ve Mellon bu konuda öncü çalışmalar yapmışlardır. Bunları diğer çalışmalar izlemiştir.

Demir Çelik endüstrisi ve madencilik için değişik çalışmalar yapılmıştır. Demir çelik endüstrisinde üretim planlamasında ve yüksek fırınlara hammaddelerin yüklenmesiyle ilgili bir doğrusal programlama modelini T. Fabian geliştirmiştir. Ayrıca değişik demir cevherlerinin değerlendirilmesinde ve düşük kaliteli cevherlerin peletlenmesinde doğrusal programlamadan yararlanılmıştır.

DP – Karışım Problemleri

Dökümhanelerde kupol ya da endüksiyon ocağına malzeme yüklenmesinde değişik kombinasyonlar söz konusudur. Minimum maliyette, istenen kalitede alaşımlı döküm elde etmek için, doğrusal programlamadan yararlanılarak, değişik çalışmalar yapılmıştır.

Görüleceği üzere doğrusal programlama tekniği başta üretim planlama problemleri olmak üzere endüstride ve diğer sahalarda oldukça geniş bir uygulama alanına sahiptir. Çok değişik sorunlara uygulanmış ve büyük tasarruflar sağlanmıştır.

DP – Karışım Problemleri

Her modelde olduğu gibi doğrusal programlama modelleri de bazı varsayımlardan hareket edilerek düzenlenmektedir. Kuşkusuz sonuçların geçerliliği modelin hazırlanmasında gerekli bilgilerin tutarlılığına bağlıdır. Varılan sonuçlar mutlak ve değişmez veriler değildir. Verilerin toplandığı dönemle uygulanacağı dönemin farklı olması ya da istenen doğrulukta bilgi bulunamaması gibi nedenlerle kesin sonuçlara varılamayabilmektedir. Ancak soruna yaklaşımın nasıl yapılacağı belirlendiği için yeni verilerle gerçek durumu tespit etmek her zaman mümkündür.

Uygulama çalışmalarının işletme için en önemli yanı her şeyden önce mevcut durumu gözler önüne sermesidir. Ayrıca, üretim, pazarlama ve satın alma gibi işletme politikalarının yönlendirilmesinde, yönetici elinde güçlü bir araç olmasıdır.

DP – Grafik Çözüm – Örnek

Doğrusal Programlamada 2 karar değişkeni olduğu durumlarda grafik yöntem ile çözüm yapılabilir.

Örneğin, aşağıda verilen matematiksel modeli grafik yöntem ile çözelim.

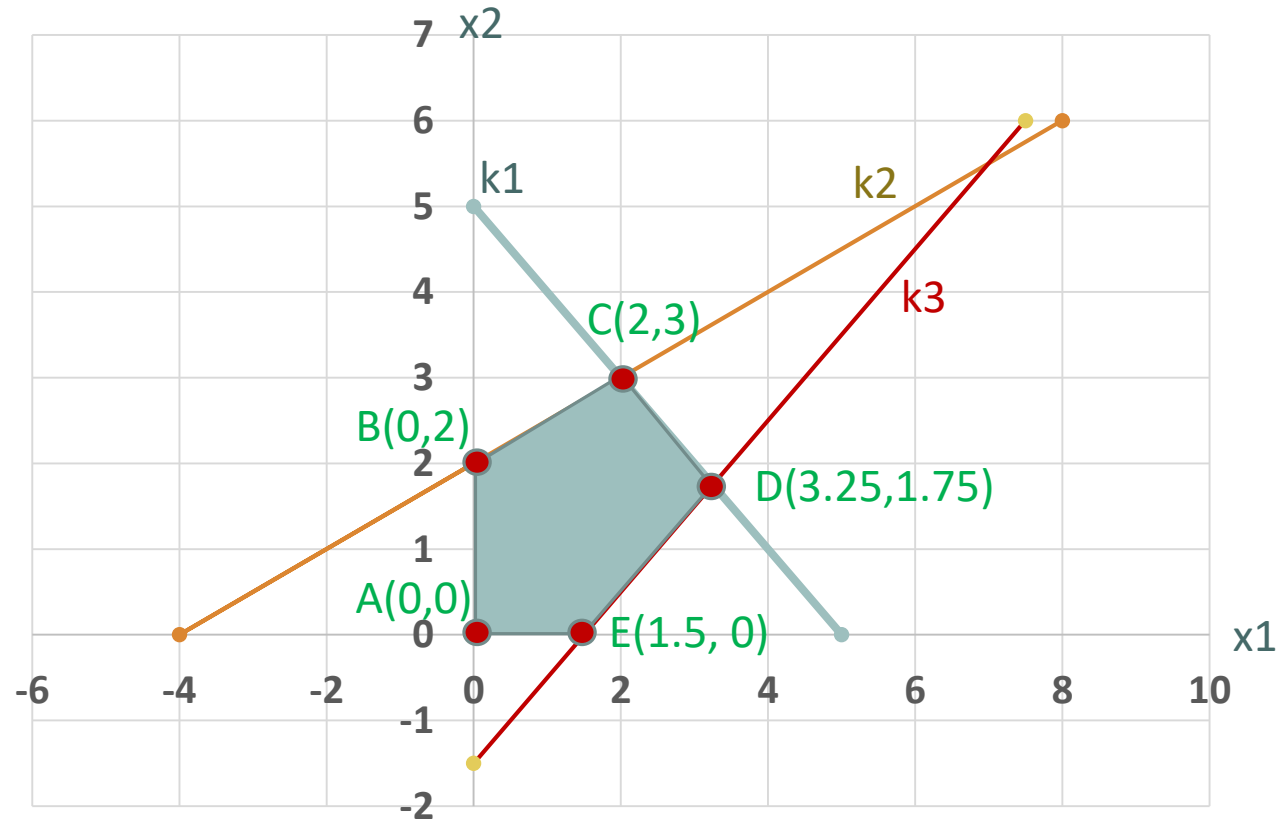
$$\text{Enb } Z = 5x_1 + 3x_2$$

$$x_1 + x_2 \leq 5 \quad (k1)$$

$$-x_1 + 2x_2 \leq 4 \quad (k2)$$

$$2x_1 - 2x_2 \leq 3 \quad (k3)$$

$$x_1, x_2 \geq 0$$

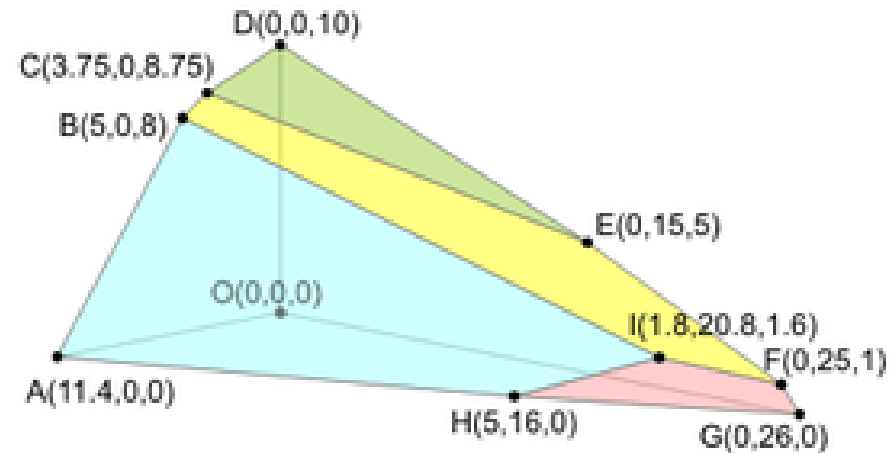


Nokta	Z Değeri
A	0
B	6
C	19
D	21.5*
E	7.5

DP – Grafik Çözüm – Örnek

Peki ikiden fazla karar değişkeni olsaydı ne olurdu? Üç adet karar değişkeninden oluşan aşağıdaki örneği ve grafik çözümünü inceleyelim.

$$\begin{array}{llll} \text{Maximize} & P = & 20x_1 & + & 10x_2 & + & 15x_3 \\ \text{Subject to:} & & 3x_1 & + & 2x_2 & + & 5x_3 \leq 55 \\ & & 2x_1 & + & x_2 & + & x_3 \leq 26 \\ & & x_1 & + & x_2 & + & 3x_3 \leq 30 \\ & & 5x_1 & + & 2x_2 & + & 4x_3 \leq 57 \\ & & x_1 & , & x_2 & , & x_3 \geq 0 \end{array}$$

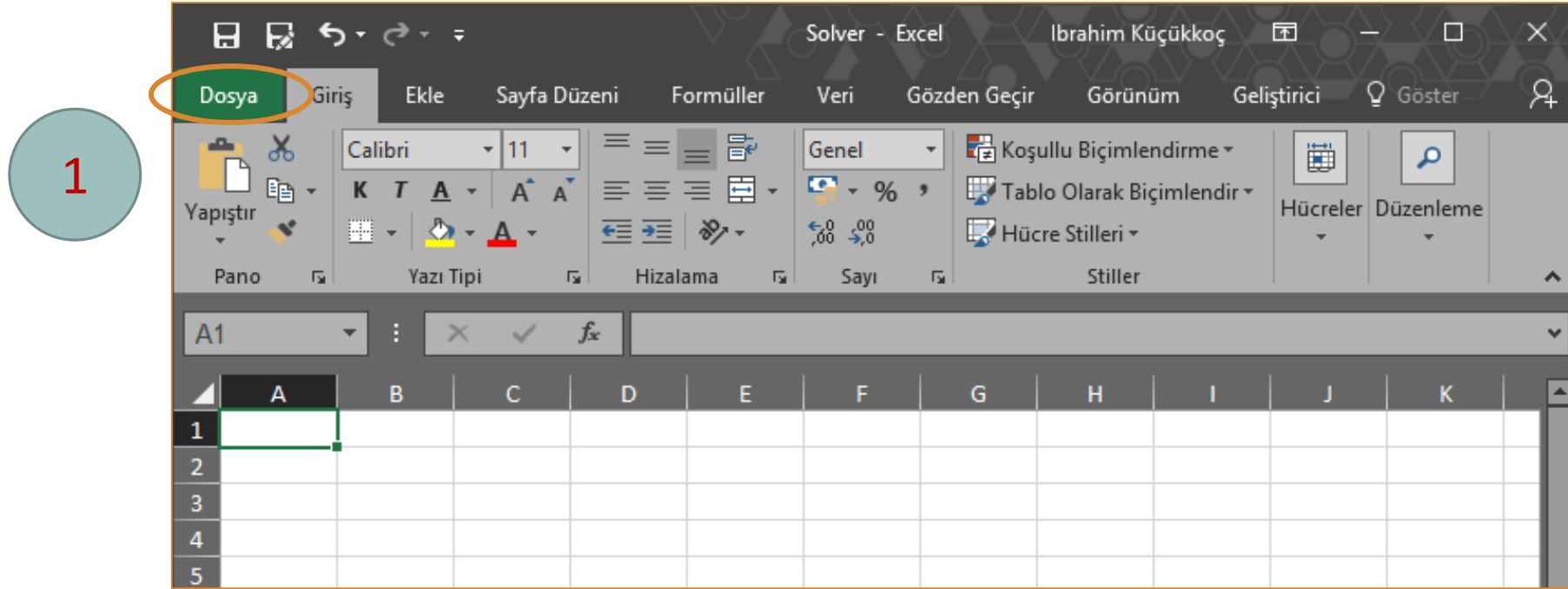


Problem Çözmede Excel Solver Kullanımı



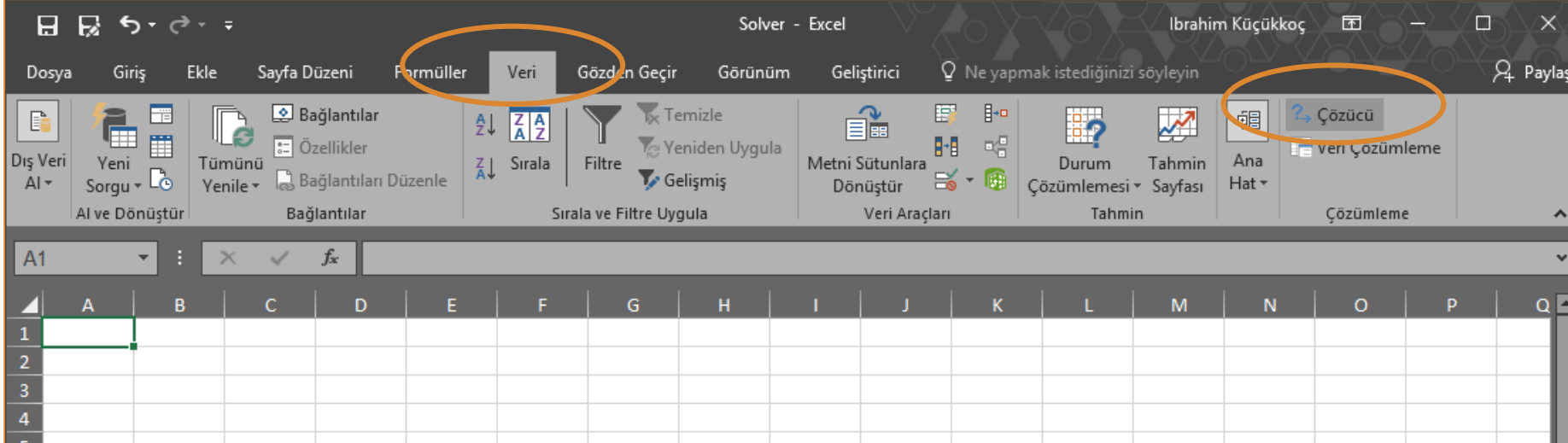
Excel Solver Kurulumu ve Kullanımı

Solver eklentisi, Excel ile birlikte gelen bir araçtır fakat kullanabilmek için öncelikle Excel'de **Dosya -> Seçenekler-> Eklentiler** yolu izlenerek **Çözücü (Solver) Eklentisi** etkinleştirilmelidir. Daha sonra Excel kapatılıp yeniden açılmalıdır.



Excel Solver Kurulumu ve Kullanımı

Excel'i kapatıp yeniden başlattıktan sonra, **Veri** menüsüne eklenmiş olan ve kullanıma hazır bulunan **Çözücü** eklentisini görebilirsiniz.



Excel Solver Kurulumu ve Kullanımı

Çözücü Parametreleri

Hedef Ayarla: 1

2 Hedef: ☒ En Büyük ☐ En Küçük ☐ Değeri:

Değişken Hücreleri Değiştirerek: 3

Kısıtlamalara Bağlıdır: 4

☒ Kısıtlanmamış Değişkenleri Pozitif Yap

Çözme Yöntemi Seçin: Doğrusal Olmayan GRG

Seçenekler

Çözüm Yöntemi

Düğüün doğrusal olmayan Çözücü Problemleri için GRG Doğrusal Olmayan altyapısını seçin.
Doğrusal Çözücü Problemleri için Basit LP altyapısını seçin ve düğüün olmayan Çözücü problemleri için Açılım altyapısını seçin.

Yardım Çöz Kapat

Çözücü'ye tıkladığınızda ekrana gelen panel sizden aşağıdaki bilgileri isteyecektir:

- 1: Amaç fonksiyonu değeri bulduğu hücre (Bkz: Hedef Ayarla)
- 2: Amaç fonksiyonu stratejisi -maksimizasyon, minimizasyon, hedef değeri- (Bkz: Hedef)
- 3: Karar değişkenlerinin bulunduğu hücreler (Bkz: Değişken Hücreleri Değiştirerek)
- 4: Kısıtlar (Bkz: Kısıtlamalara Bağlıdır)

Excel Solver Kurulumu ve Kullanımı

Excel Solver Kullanarak Problem Çözme Adımları:

Bir optimizasyon probleminin Excel Solver kullanılarak çözülmesi için sırasıyla aşağıdaki adımlar takip edilir.

- Problem verisi Excel üzerinde mantıklı bir şekilde düzenlenir.
- Modeldeki her bir karar değişkeninin alacağı değeri tutmak üzere bir Excel hücresi belirlenir.
- Bir hücreye, modelin amaç fonksiyonu değerini hesaplayacak olan formülasyon yazılır.
- Her bir kısıt için sol taraf değerlerini hesaplayacak olan bir formülasyon ilgili hücrelere yazılır.
- Excel Solver paneli açılarak karar değişkenleri, amaç fonksiyonu, kısıtlar ve sınırlamalar ilgili kutucuklarda tanımlanır.
- Solver çalıştırılır.

Excel Solver-Örnek 1: Ahşap Panel Üretimi

Dört tip ahşap panelin (Tahoe, Pacific, Savannah, Aspen) üretildiği bir sistemi ele alalım. Her ahşap panel, çam ve meşe talaşlarının farklı miktarlarda karıştırılıp yapıştırılması ve preslenmesi sonucu oluşmaktadır.

Aşağıdaki tablo, değişik panel tipleri için bir palet panel üretiminde kullanılması gereken kaynak miktarlarını göstermektedir.

Her Bir Palet İçin Gerekli Kaynak Miktarı				
Kaynak/Panel	Tahoe	Pacific	Savannah	Aspen
Yapıştırıcı (litre)	50	50	100	50
Sıkıştırma (saat)	5	15	10	5
Çam Talaşı (kg)	500	400	300	200
Meşe Talaşı (kg)	500	750	250	500

Excel Solver-Örnek 1

Eldeki (mevcut) kaynak miktarları şu şekildedir:

5800 litre yapıştırıcı

730 saat sıkıştırma kapasitesi

29200 kg çam talaşı

60500 kg meşe talaşı

Her bir palet Tahoe, Pacific, Savannah ve Aspen paneli sırasıyla \$450, \$1150, \$800 ve \$400'a satılmaktadır.

Firma karını maksimize etmek için her bir panel tipinden kaç palet üretmelidir?

Excel Solver-Örnek 1

Öncelikle problem matematiksel model olarak ifade edilmelidir.

Karar değişkenleri Tahoe, Pacific, Savannah ve Aspen modellerinin üretim miktarlarıdır.

Bunlar sırasıyla X_1 , X_2 , X_3 ve X_4 olarak düşünülürse amaç fonksiyonu aşağıdaki şekilde yazılabilir:

$$\text{Max: } 450X_1 + 1150X_2 + 800X_3 + 400X_4$$

Kaynak kullanımlarına ilişkin kısıtlar aşağıdaki şekilde ifade edilebilir:

$$50 X_1 + 50 X_2 + 100 X_3 + 50 X_4 \leq 5800 \text{ (yapıştırıcı)}$$

$$5 X_1 + 15 X_2 + 10 X_3 + 5 X_4 \leq 730 \text{ (sıkıştırma)}$$

$$500 X_1 + 400 X_2 + 300 X_3 + 200 X_4 \leq 29200 \text{ (çam)}$$

$$500 X_1 + 750 X_2 + 250 X_3 + 500 X_4 \leq 60500 \text{ (meşe)}$$

Herhangi bir panelin üretim miktarı 0'dan küçük olamayacağı için:

$$X_1, X_2, X_3, X_4 \geq 0$$

Excel Solver-Örnek 1

- a) Öncelikle problem verisi Excel’de uygun ve mantıksal bir formatta yazılmalıdır. Bu aşamada henüz bir formül kullanılmamıştır:

	Panel Tipi					
	Tahoe	Pacific	Savannah	Aspen		
Kazanç	\$450	\$1,150	\$800	\$400		
	Her Bir Palet İçin Gerekli Kaynak Miktarı				Mevcut	
Yapıştırıcı	50	50	100	50	5,800	litre
Sıkıştırma	5	15	10	5	730	saat
Çam Talaşı	500	400	300	200	29,200	kg
Meşe Talaşı	500	750	250	500	60,500	kg

Excel Solver-Örnek 1

b) Modeldeki karar değişkenlerinin alacağı değerleri tutmak üzere hücreler oluşturulur:

	Panel Tipi					
	Tahoe	Pacific	Savannah	Aspen		
Pallet Sayısı	0	0	0	0		
Kazanç	\$450	\$1,150	\$800	\$400		
	Her Bir Palet İçin Gerekli Kaynak Miktarı				Mevcut	
Yapıştırıcı	50	50	100	50	5,800	litre
Sıkıştırma	5	15	10	5	730	saat
Çam Talaşı	500	400	300	200	29,200	kg
Meşe Talaşı	500	750	250	500	60,500	kg

Excel Solver-Örnek 1

- c) Bir hücreye, modelin amaç fonksiyonu değerini hesaplayacak olan formülasyon yazılır.

Burada toplam kazanç hesabı için '**TOPLA.ÇARPIM**' formülü kullanılır

=TOPLA.ÇARPIM(C19:F19,C18:F18)

Bu formül aşağıdaki işlemi yerine getirir:

$$450X_1 + 1150X_2 + 800X_3 + 400X_4$$

	Panel Tipi					
	Tahoe	Pacific	Savannah	Aspen		
Pallet Sayısı	0	0	0	0	Toplam Kazanç	
Kazanç	\$450	\$1,150	\$800	\$400	\$0	
	Her Bir Palet İçin Gerekli Kaynak Miktarı				Mevcut	
Yapıştırıcı	50	50	100	50	5,800	litre
Sıkıştırma	5	15	10	5	730	saat
Çam Talaşı	500	400	300	200	29,200	kg
Meşe Talaşı	500	750	250	500	60,500	kg

Excel Solver-Örnek 1

- d) Her bir kısıt için sol taraf değerlerini hesaplayacak olan bir formülasyon ilgili hücrelere yazılır. Resimde belirtildiği üzere 'Kullanılan' başlığı altında belirtilen hücreler bu amaçla oluşturulmuştur. Örneğin kullanılan yapıştırıcı miktarını hesaplayan

$$50 X_1 + 50 X_2 + 100 X_3 + 50 X_4 \leq 5800 \text{ formülü}$$

=TOPLA.ÇARPIM(C22:F22,\$C\$18:\$F\$18)

olarak yazılmıştır.

NOT: Şu anda karar değişkeni değerleri '0' olduğu için kullanılan kaynak miktarı değerleri de '0' olarak görünmektedir.

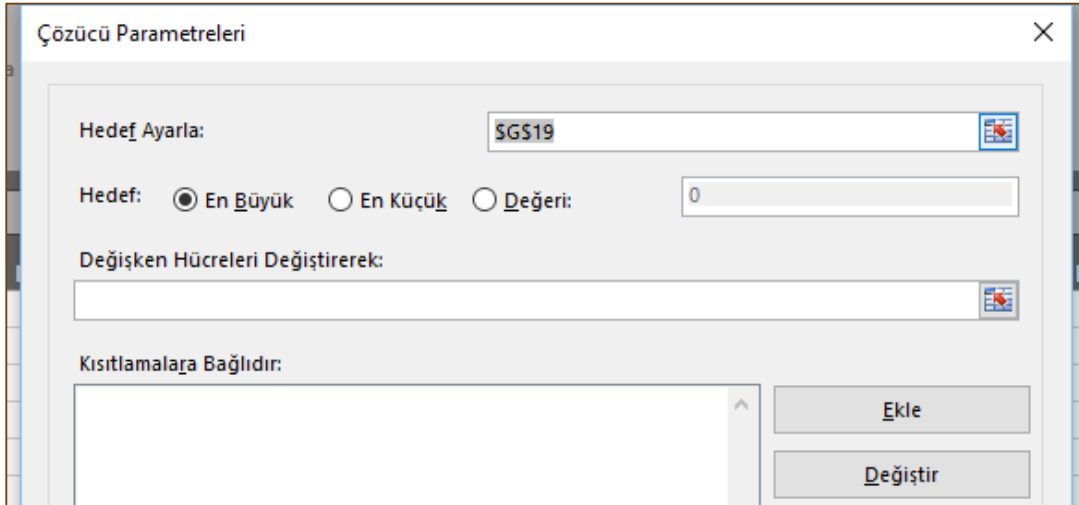
	Panel Tipi						
	Tahoe	Pacific	Savannah	Aspen			
Pallet Sayısı	0	0	0	0	Toplam Kazanç		
Kazanç	\$450	\$1,150	\$800	\$400	\$0		
	Her Bir Palet İçin Gerekli Kaynak Miktarı				Kullanılan	Mevcut	
Yapıştırıcı	50	50	100	50	0	5,800	litre
Sıkıştırma	5	15	10	5	0	730	saat
Çam Talaşı	500	400	300	200	0	29,200	kg
Meşe Talaşı	500	750	250	500	0	60,500	kg

Excel Solver-Örnek 1

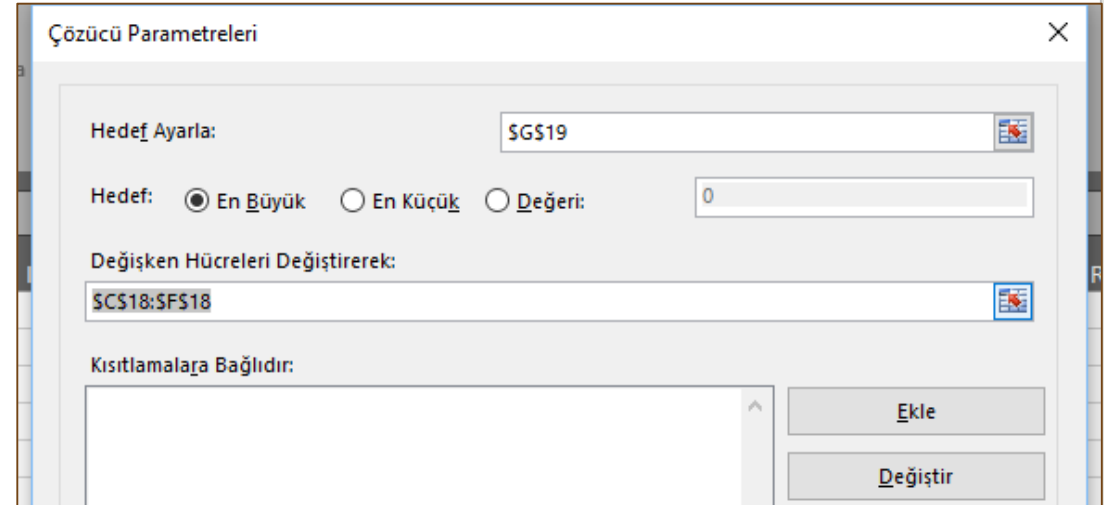
- e) Excel Solver paneli açılarak karar değişkenleri, amaç fonksiyonu, kısıtlar ve sınırlamalar ilgili kutucuklarda tanımlanır.

*Hedef Ayarla: \$G\$19
Hedef: En Büyük*

*Değişken Hücreleri Değiştirerek:
\$C\$18:\$F\$18*



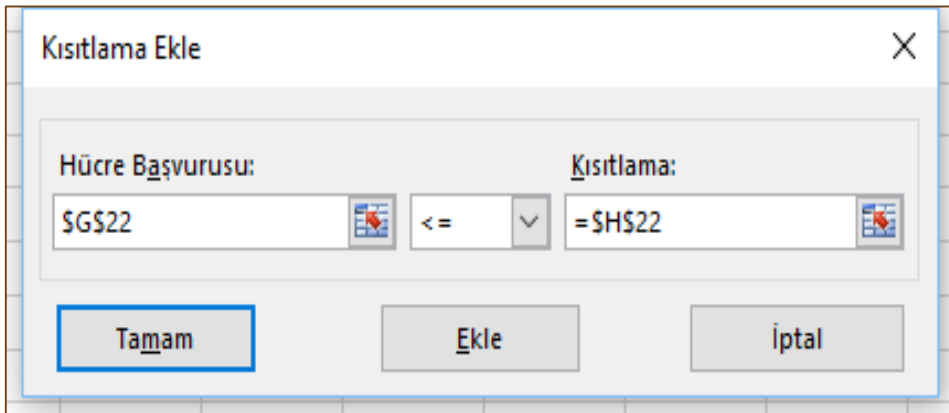
The image shows the 'Çözücü Parametreleri' (Solver Parameters) dialog box in Excel. The 'Hedef Ayarla:' (Set Objective) field contains '\$G\$19'. The 'Hedef:' (To: Of) section has three radio buttons: 'En Büyük' (Largest Of The) is selected, 'En Küçük' (Smallest Of The) is unselected, and 'Değeri:' (Value Of) is unselected with a text box containing '0'. The 'Değişken Hücreleri Değiştirerek:' (By Changing Variable Cells) field is empty. The 'Kısıtlamalara Bağlıdır:' (Subject to the Constraints) section is empty. There are 'Ekle' (Add) and 'Değiştir' (Change) buttons at the bottom right.



The image shows the 'Çözücü Parametreleri' (Solver Parameters) dialog box in Excel. The 'Hedef Ayarla:' (Set Objective) field contains '\$G\$19'. The 'Hedef:' (To: Of) section has three radio buttons: 'En Büyük' (Largest Of The) is selected, 'En Küçük' (Smallest Of The) is unselected, and 'Değeri:' (Value Of) is unselected with a text box containing '0'. The 'Değişken Hücreleri Değiştirerek:' (By Changing Variable Cells) field contains '\$C\$18:\$F\$18'. The 'Kısıtlamalara Bağlıdır:' (Subject to the Constraints) section is empty. There are 'Ekle' (Add) and 'Değiştir' (Change) buttons at the bottom right.

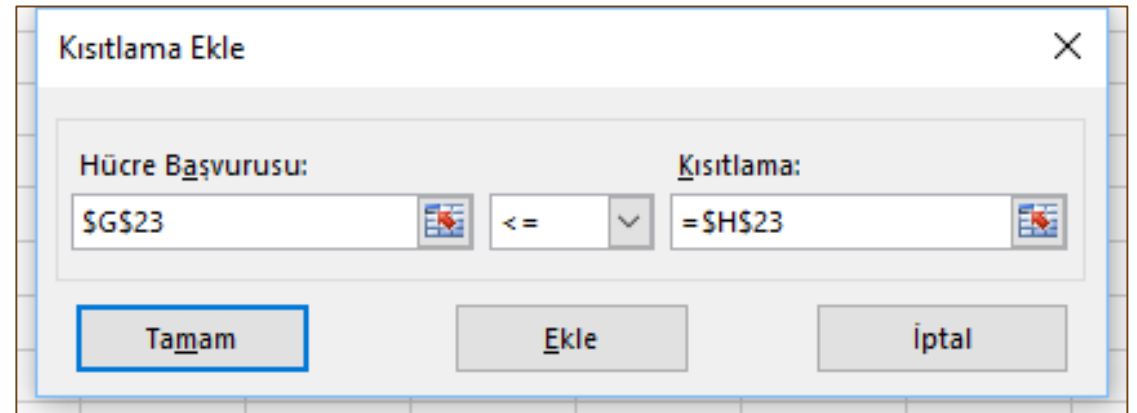
Excel Solver-Örnek 1

Yapıştırıcı kısıtı: Kullanılan miktar mevcut miktardan küçük veya eşit olmalıdır:
 $\$G\$22 \leq \$H\22



The 'Kısıtlama Ekle' (Add Constraint) dialog box in Excel. The 'Hücre Başvurusu:' (Cell Reference) field contains '\$G\$22'. The 'Kısıtlama:' (Constraint) field contains '\$H\$22'. The comparison operator is set to '<=' (less than or equal to). The 'Tamam' (OK) button is highlighted with a blue border.

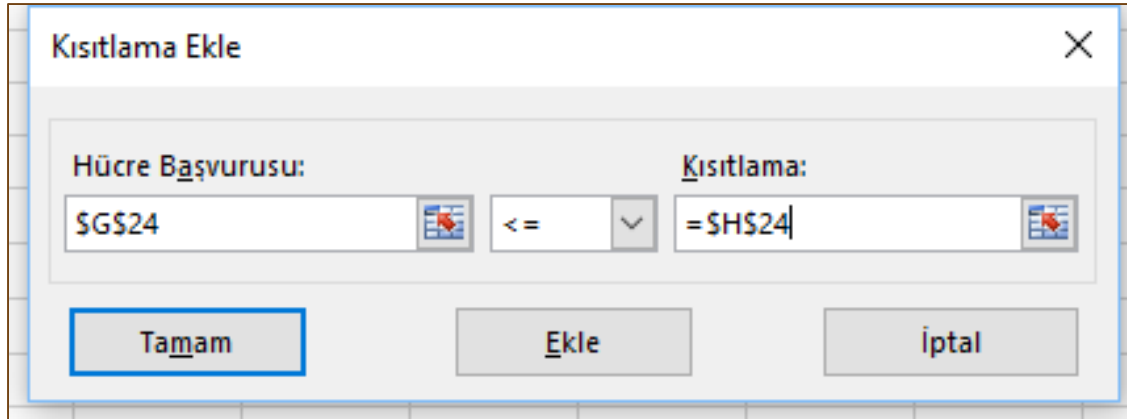
Sıkıştırma kısıtı: Kullanılan miktar mevcut miktardan küçük veya eşit olmalıdır:
 $\$G\$23 \leq \$H\23



The 'Kısıtlama Ekle' (Add Constraint) dialog box in Excel. The 'Hücre Başvurusu:' (Cell Reference) field contains '\$G\$23'. The 'Kısıtlama:' (Constraint) field contains '\$H\$23'. The comparison operator is set to '<=' (less than or equal to). The 'Tamam' (OK) button is highlighted with a blue border.

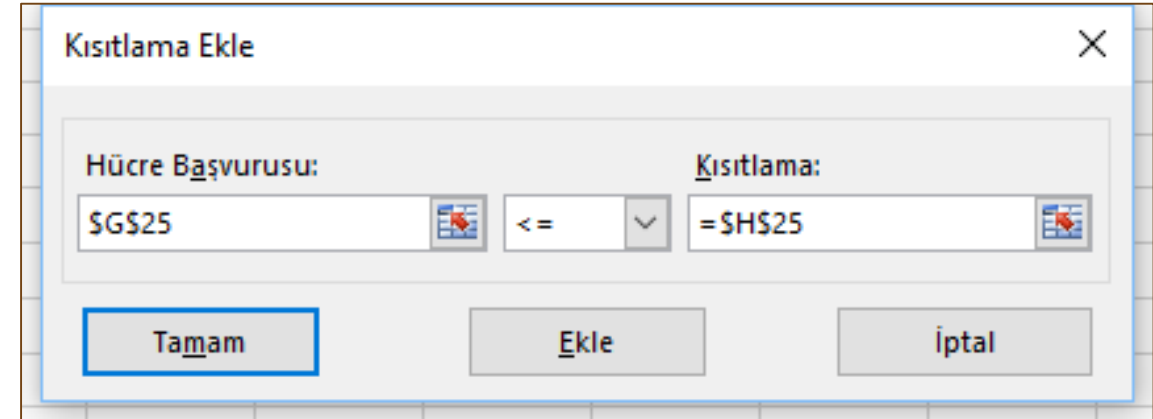
Excel Solver-Örnek 1

Çam talaşı kısıtı: Kullanılan miktar mevcut miktardan küçük veya eşit olmalıdır:
 $\$G\$24 \leq \$H\24



The image shows the 'Add Constraint' (Kısıtlama Ekle) dialog box in Excel. The 'Cell Reference' (Hücre Başvurusu) field contains '\$G\$24'. The 'Constraint' (Kısıtlama) field contains '\$H\$24'. The comparison operator is set to '<=' (less than or equal to). The 'Add' (Ekle) button is highlighted with a blue border. The 'Complete' (Tamam) and 'Cancel' (İptal) buttons are also visible.

Meşe talaşı kısıtı: Kullanılan miktar mevcut miktardan küçük veya eşit olmalıdır:
 $\$G\$25 \leq \$H\25

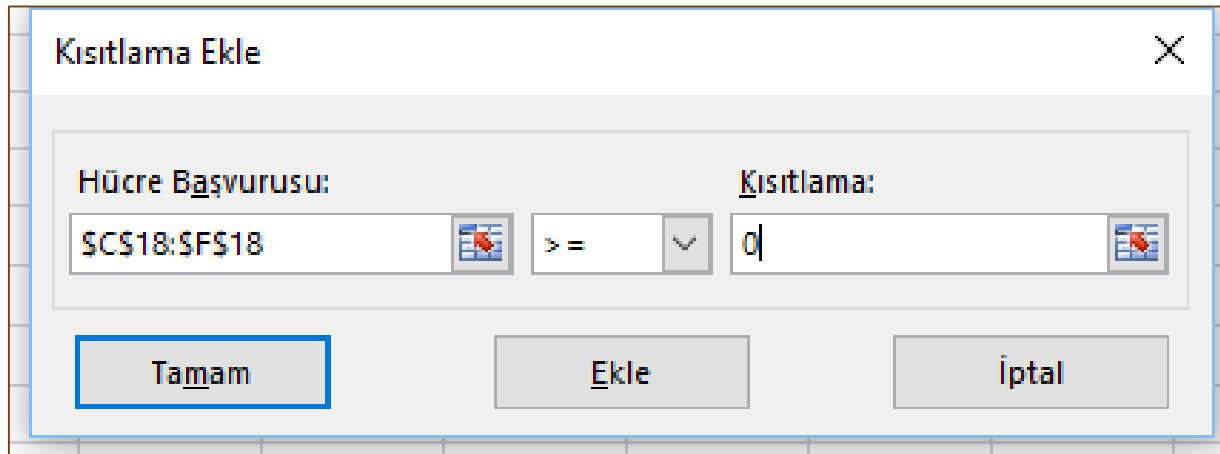


The image shows the 'Add Constraint' (Kısıtlama Ekle) dialog box in Excel. The 'Cell Reference' (Hücre Başvurusu) field contains '\$G\$25'. The 'Constraint' (Kısıtlama) field contains '\$H\$25'. The comparison operator is set to '<=' (less than or equal to). The 'Add' (Ekle) button is highlighted with a blue border. The 'Complete' (Tamam) and 'Cancel' (İptal) buttons are also visible.

Excel Solver-Örnek 1

Herhangi bir panelin üretim miktarı 0'dan küçük olamayacağı için tüm karar değişkenleri sıfırdan büyük veya eşit olmalıdır:

$$\$C\$18:\$F\$18 \geq 0$$



The image shows the 'Add Constraint' (Kısıtlama Ekle) dialog box in Excel. The 'Cell Reference' (Hücre Başvurusu) field contains '\$C\$18:\$F\$18'. The 'Constraint' (Kısıtlama) field contains '0'. The comparison operator is set to 'greater than or equal to' (>=). The 'Add' (Ekle) button is highlighted with a blue border. The 'OK' (Tamam) and 'Cancel' (İptal) buttons are also visible.

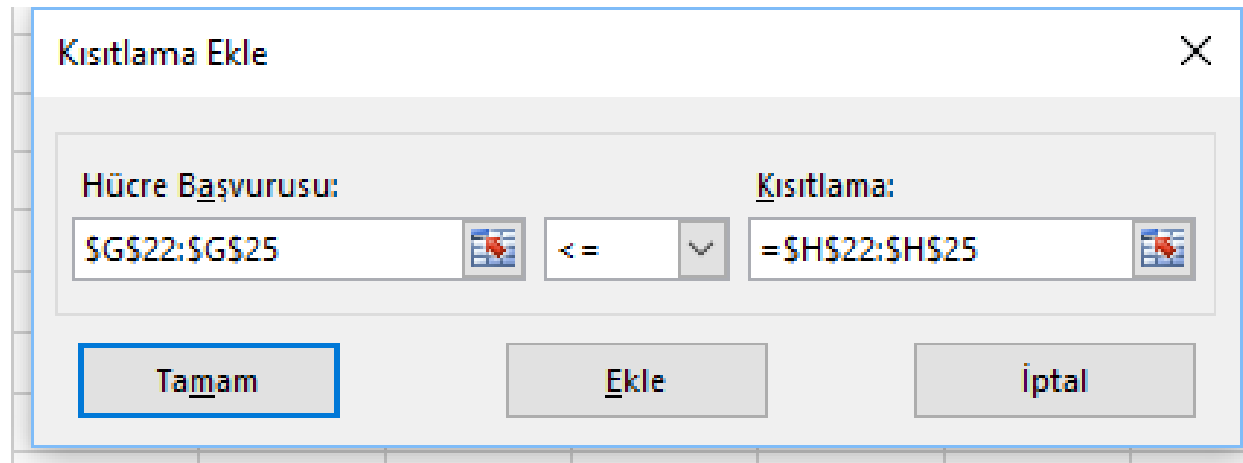
Hücre Başvurusu:	Kısıtlama:
\$C\$18:\$F\$18	0

>=

Tamam Ekle İptal

Excel Solver-Örnek 1

NOT: Yapıştırma, sıkıştırma, çam talaşı ve meşe talaşı kısıtlarını tek tek girmektense, aşağıdaki şekilde toplu olarak da girilebilir. Bu durumda elde edilecek olan sonuç değişmeyecektir.
\$G\$22:\$G\$25 <= \$H\$22:\$H\$25).



The image shows the 'Kısıtlama Ekle' (Add Constraint) dialog box in Excel. The 'Hücre Başvurusu:' (Cell Reference) field contains '\$G\$22:\$G\$25'. The 'Kısıtlama:' (Constraint) field contains '\$H\$22:\$H\$25'. The comparison operator is set to '<=' (less than or equal to). There are three buttons at the bottom: 'Tamam' (OK), 'Ekle' (Add), and 'İptal' (Cancel). The 'Tamam' button is highlighted with a blue border.

Excel Solver-Örnek 1

- f) Çöz butonuna tıklanarak Solver çalıştırılır.

Çözücü Parametreleri

Hedef Ayarla:

Hedef: ☒ En Büyük ☐ En Küçük ☐ Değeri:

Değişken Hücreleri Değiştirerek:

Kısıtlamalara Bağlıdır:

☒ Kısıtlanmamış Değişkenleri Pozitif Yap

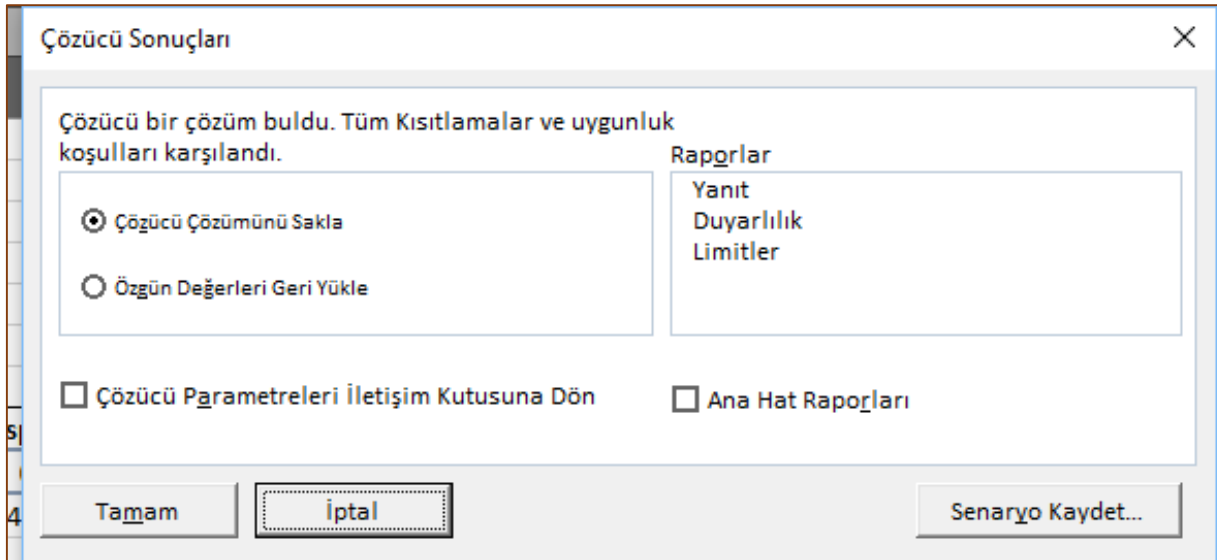
Çözme Yöntemi:

Çözüm Yöntemi

Düzgün doğrusal olmayan Çözücü Problemleri için GRG Doğrusal Olmayan altyapısını seçin.
Doğrusal Çözücü Problemleri için Basit LP altyapısını seçin ve düzgün olmayan Çözücü problemleri için Açılım altyapısını seçin.

Excel Solver-Örnek 1

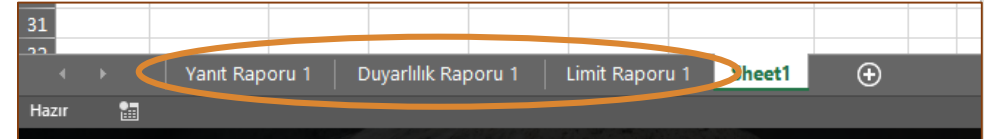
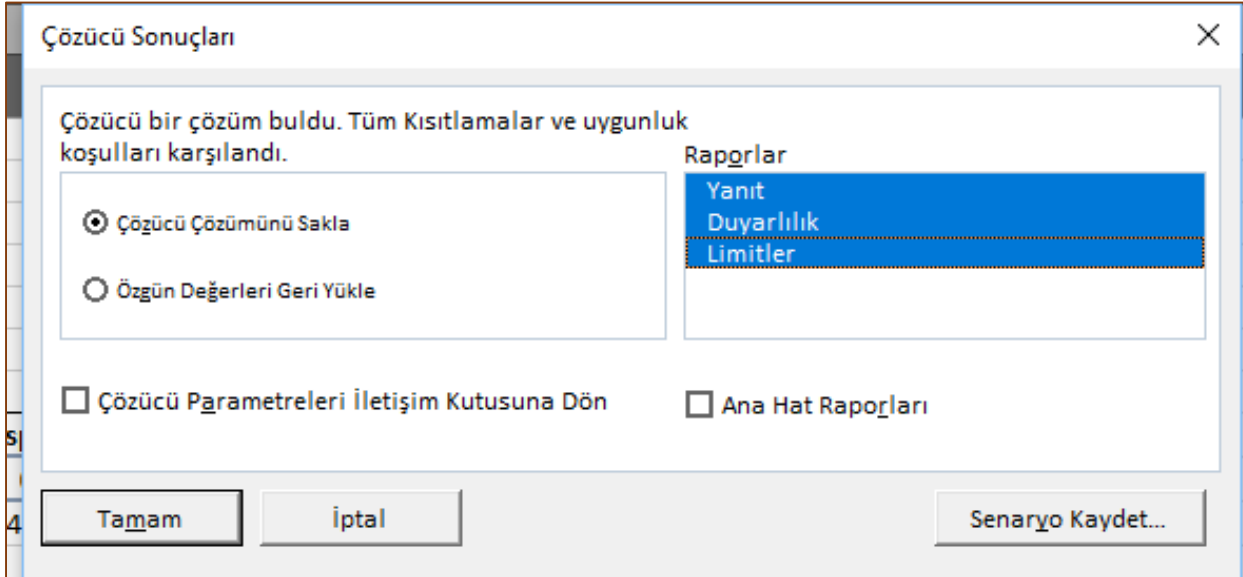
Solver bir süre çalıştıktan sonra çözümü buldu ve bu çözümün tüm kısıtları ve uygunluk koşullarını sağladığını belirtti:



Aşağıdaki resimden görüleceği üzere, bulunan çözüme göre karar değişkenlerinin ve amaç fonksiyonunun değerleri ilgili hücrelere yazılmıştır. Bu sonuca göre Tahoe, Pacific ve Savannah modellerinden 23, 15 ve 39 palet üretilmesi maksimum kazancı (\$58,800) sağlayacaktır. Yapıştırıcı, sıkıştırma ve çam talaşı kapasitelerinin tamamı kullanılmış, meşe talaşı kısıtının ise 32500 kg'ı kullanılmıştır.

	Tahoe	Pacific	Savannah	Aspen			
Pallet Sayısı	23	15	39	0	Toplam Kazanç		
Kazanç	\$450	\$1,150	\$800	\$400	\$58,800		
	Her Bir Palet İçin Gerekli Kaynak Miktarı				Kullanılan	Mevcut	
Yapıştırıcı	50	50	100	50	5,800	5,800	litre
Sıkıştırma	5	15	10	5	730	730	saat
Çam Talaşı	500	400	300	200	29,200	29,200	kg
Meşe Talaşı	500	750	250	500	32,500	60,500	kg

İstenen raporlar (Yanıt, Duyarlılık, Limit) seçilerek Tamam butonuna tıklanırsa, Excel'de yeni sayfalarda raporlar oluşturulur.



A	B	C	D	E	F	G	H
			Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
\$B\$4	Karar Değ. Tahoe		23	0	450	300	50
\$C\$4	Karar Değ. Pacific		15	0	1150	42,8571429	714,285714
\$D\$4	Karar Değ. Savannah		39	0	800	100	16,6666667
\$E\$4	Karar Değ. Aspen		0	-7,14285714	400	7,14285714	1E+30

Constraints

		Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
\$G\$8	Yapıştırıcı Kullanılan	5800	0,428571429	5800	1500	2481,81818
\$G\$9	Sıkıştırma Kullanılan	730	71,42857143	730	322	150
\$G\$10	Çam Talaşı Kullanılan	29200	0,142857143	29200	26133,3333	8050
\$G\$11	Meşe Talaşı Kullanılan	32500	0	60500	1E+30	28000

Excel Solver-Örnek 2: Bilgisayar Firması (Ürün Karışımı)

Problem:

ABC firması, standart ve lüks bilgisayar olarak adlandırdığı iki tip bilgisayar üretmektedir. Standart bilgisayarlar için normal kapasiteli, lüks bilgisayarlar içinse yüksek kapasiteli sabit disk (SD) kullanılmaktadır. Ayrıca standart bilgisayarlarda bir adet, lüks bilgisayarlarda iki adet RAM vardır. Firma, standart bilgisayar başına \$30, lüks bilgisayar başına \$50 kar etmektedir. Bir ayda en çok 60 adet normal kapasiteli SD, 50 adet yüksek kapasiteli SD ve 120 adet RAM kullanılabilir. Firma, stok değerlerini aşmayacak şekilde iki tip bilgisayardan ayda kaç adet üretmesi durumunda kar maksimize edilecektir?

Excel Solver-Örnek 2

Model:

Firmanın aylık standart bilgisayar üretim miktarı X_s , aylık lüks bilgisayar üretimi X_l ile gösterilirse; amaç karı maksimize etmek olduğundan amaç fonksiyonu:

$$\text{Max: } 30X_s + 50X_l$$

Kısıtlar:

$$X_s \leq 60$$

$$X_l \leq 50$$

$$X_s + 2X_l \leq 120$$

$$X_s, X_l \geq 0$$

Excel Solver-Örnek 2

Karar Değişkenleri

Bu örnekte üretilecek ürünlerin optimum miktarının belirlenmesi isteniyor.

Dolayısıyla, C10:D10 hücreleri karar değişkenleridir.

Değişken Sınırları

Negatif sayıda ürün üretilmeyeceği için değişkenler en düşük 0 değerini alabilirler. Bu durum $C10:D10 \geq 0$ olarak Çözücüde kısıt olarak belirtilmiştir.

Alternatif olarak Çözücüde "Kısıtlanmamış Değişkenleri Pozitif Yap" seçeneği de seçilebilir.

Amaç Fonksiyonu

Modeldeki amacımız karı maksimize etmektir. Bu amaçla, modelde kullanmak üzere F11 hücrelerini belirliyoruz.

Kısıtlar

Örnekte, kullanılacak malzeme veya kapasite kısıtlarını içeren dört grup kısıt bulunmaktadır.

Çözücüde kısıt tanımlamaları alanında $E16:E18 \leq F16:F18$ şeklinde eklenecek olan eşitsizlik bu kısıtları modele dahil edecektir.

	Ürün Modeli			
	Standart	Lüks		
Üretim miktarı	0	0		
Birim kazanç	30	50		
	Gerekli Miktar		Kullanılan	Stok
Bileşenler	Standart	Lüks		
Normal kap. SD	1	0	0	60
Yüksek kap. SD	0	1	0	50
RAM	1	2	0	120

Toplam Kazanç

0

Excel Solver-Örnek 2

Çözücü arayüzü

Çözücü Parametreleri

Hedef Ayarla:

Hedef: ☒ En Büyük ☐ En Küçük ☐ Değeri:

Değişken Hücreleri Değiştirerek:

Kısıtlamalara Bağlıdır:

☒ Kısıtlanmamış Değişkenleri Pozitif Yap

Çözme Yöntemi Seçin:

Çözüm Yöntemi

Düzgün doğrusal olmayan Çözücü Problemleri için GRG Doğrusal Olmayan altyapısını seçin.
Doğrusal Çözücü Problemleri için Basit LP altyapısını seçin ve düzgün olmayan Çözücü problemleri için Açılım altyapısını seçin.

Karar Değişkenleri

Bu örnekte üretilecek ürünlerin optimum miktarının belirlenmesi isteniyor.

Dolayısıyla, C10:D10 hücreleri karar değişkenleridir.

Değişken Sınırları

Negatif sayıda ürün üretilmeyeceği için değişkenler en düşük 0 değerini alabilirler. Bu durum C10:D10 ≥ 0 olarak Çözücüde kısıt olarak belirtilmiştir.

Alternatif olarak Çözücüde "Kısıtlanmamış Değişkenleri Pozitif Yap" seçeneği de seçilebilir.

Amaç Fonksiyonu

Modeldeki amacımız karı maksimize etmektir. Bu amaçla, modelde kullanmak üzere F11 hücresini belirliyoruz.

Kısıtlar

Örnekte, kullanılacak malzeme veya kapasite kısıtlarını içeren dört grup kısıt bulunmaktadır.

Çözücüde kısıt tanımlamaları alanında E16:E18 $\leq$ F16:F18 şeklinde eklenecek olan eşitsizlik bu kısıtları modele dahil edecektir.

	Ürün Modeli			
	Standart	Lüks		
Üretim miktarı	60	30		Toplam Kazanç
Birim kazanç	30	50		3300
	Gerekli Miktar		Kullanılan	Stok
Bileşenler	Standart	Lüks		
Normal kap. SD	1	0	60	60
Yüksek kap. SD	0	1	30	50
RAM	1	2	120	120

Excel Solver-Örnek 3: Elektronik Sirketi (Ürün Karışımı)

Firmanız, LCD TV, Muzik Calar ve Hoparlor üretimi yapmaktadır. Bu ürünlerin üretiminde, Govde, LCD Ekran, Hoparlor, Guc unitesi ve Cesitli Elektronik Parcalari "ortak" olarak kullanmaktadır. Kullanilacak parcalarin tedarigi sınırlı iken karı maksimize edecek optimum üretim miktarlarını bulunuz.

Part Requirements by Product				Malzeme Stok Miktarlari
Parca Adi	LCD TV	Muzik Calar	Hoparlor	Mevcut
Govde	1	1	0	450
LCDEkran	1	0	0	250
Hoparlor	2	2	1	800
Guc Unitesi	1	1	0	450
Elektronik Parca	2	1	1	600
Birim Kar	\$75	\$50	\$35	

Excel Solver-Örnek 3

Karar Değişkenleri

Bu örnekte üretilecek ürünlerin optimum miktarının belirlenmesi isteniyor. Dolayısıyla, C14:E14 hücreleri karar değişkenleridir.

Değişken Sınırları

Negatif sayıda ürün üretilmeyeceği için değişkenler en düşük 0 değerini alabilirler. Bu durum C14:E14 ≥ 0 şeklinde Çözücüde kısıt olarak belirtilmiştir. Alternatif olarak Çözücüde "Kısıtlanmamış Değişkenleri Pozitif Yap" seçeneği de seçilebilir.

Uretim Kararlari

	LCD TV	Muzik Calar	Hoparlör
Uretim Miktarı	0	0	0

Part Requirements by Product

Parca Adi	LCD TV	Muzik Calar	Hoparlör
Govde	1	1	0
LCDEkran	1	0	0
Hoparlör	2	2	1
Guc Unitesi	1	1	0
Elektronik Parca	2	1	1
Birim Kar	\$75	\$50	\$35

Malzeme Stok Miktarlari

Kullanilan	Mevcut
0	450
0	250
0	800
0	450
0	600

Kısıtlar

Örnek model, kullanılacak malzeme miktarlarını sınırlayan 5 tane kısıt içermektedir. Çözücüde $Number_used \leq Number_available$ kısıtı bu durumu ifade etmektedir. Bu aynı zamanda G18:G22 $\leq$ H18:H22 anlamına gelmektedir.

Amaç Fonksiyonu

Modeldeki amacımız G24 hücresinde hesaplanan karı (Total_profit) maksimize etmektir.

\$0

Karar Değişkenleri

Bu örnekte üretilecek ürünlerin optimum miktarının belirlenmesi isteniyor. Dolayısıyla, C14:E14 hücreleri karar değişkenleridir.

Değişken Sınırları

Negatif sayıda ürün üretilemeyeceği için değişkenler en düşük 0 değerini alabilirler. Bu durum $C14:E14 \geq 0$ şeklinde Çözücüde kısıt olarak belirtilmiştir. Alternatif olarak Çözücüde "Kısıtlanmamış Değişkenleri Pozitif Yap" seçeneği de seçilebilir.

Uretim Kararlari

	LCD TV	Muzik Calar	Hoparlör
Uretim Miktari	200	200	0

Part Requirements by Product

Parca Adi	LCD TV	Muzik Calar	Hoparlör
Govde	1	1	0
LCDEkran	1	0	0
Hoparlör	2	2	1
Guc Unitesi	1	1	0
Elektronik Parca	2	1	1
Birim Kar	\$75	\$50	\$35

Malzeme Stok Miktarlari

Kullanilan	Mevcut
400	450
200	250
800	800
400	450
600	600

Kısıtlar

Örnek model, kullanılacak malzeme miktarlarını sınırlayan 5 tane kısıt içermektedir. Çözücüde $Number_used \leq Number_available$ kısıtı bu durumu ifade etmektedir. Bu aynı zamanda $G18:G22 \leq H18:H22$ anlamına gelmektedir.

Amaç Fonksiyonu

Modeldeki amacımız G24 hücresinde hesaplanan karı (Total_profit) maksimize etmektir.

\$25,000

Excel Solver-Örnek 4: Bütçe Kısıtlı Proje Seçimi

Bir şirket gelecek yıl için sekiz olası proje yatirimini degerlendirmektedir. Her bir projenin beklenen getirisi ve her bir proje için yapilmasi gereken baslangic yatirimi tabloda verilmistir. Butceyi (\$1.5 milyon) asmayarak Beklenen Toplam Getiri'yi maksimize edecek sekilde hangi projelerin yatirima dahil edilmesi gerektigine karar veriniz.

Proje No	Basarili Olursa Getirisi	Basarili Olma Sansi	Beklenen Getiri	Baslangic Yatirimi	[Beklenen Getiri] - [Yatirim]	Dahil Etme Karari
1	500000	100%	\$500.000	\$425.000	\$75.000	0
2	750000	100%	\$750.000	\$450.000	\$300.000	0
3	1000000	100%	\$1.000.000	\$550.000	\$450.000	0
4	600000	100%	\$600.000	\$300.000	\$300.000	0
5	500000	100%	\$500.000	\$150.000	\$350.000	0
6	500000	100%	\$500.000	\$250.000	\$250.000	0
7	450000	100%	\$450.000	\$350.000	\$100.000	0
8	500000	100%	\$500.000	\$325.000	\$175.000	0

Karar Degiskenleri

Karar degiskenleri I12:I19 hucrerinde verilmistir. 0-1 ikili degiskenlerdir. 1 olmasi, projenin yatirima dahil edilecegi anlamini tasir.

Degiskenler

Bu örnek deterministik olarak ele alınmıştır ve proje getirisi ve başarılı olma olasılığı açısından belirsizlik içermemektedir. Başarılı olma olasılığı tüm projeler için 100% olarak kabul edilmiştir. Degiskenler D12:D19 hucrerinde verilmistir.

Toplam Yatirim:

\$0

Butce Kisiti:

\$1.500.000

Toplam Getiri:

\$0

Kısıt

Toplam yatirimın, toplam butce olan \$1.5 Milyon'i asmaması ($G21 \leq G22$) gerekmektedir.

Amac Fonksiyonu

Amac (G24), toplam getiriyi maksimize etmektir.

Excel Solver-Örnek 5: Ürün Karışımı

Sirketiniz, LCD TV, Muzik Calar ve Hoparlor uretimi yapmaktadır. Bu urunlerin uretiminde, Govde, LCD Ekran, Hoparlor, Guc unitesi ve Cesitli Elektronik Parcalari "ortak" olarak kullanmaktadır. Kullanilacak parcalarin tedarigi sınırlı iken karı maksimize edecek optimum uretim miktarlarini bulunuz.

Karar Değişkenleri

Bu örnekte üretilecek ürünlerin optimum miktarının belirlenmesi isteniyor. Dolayısıyla, C14:E14 hücreleri karar değişkenleridir.

Değişken Sınırları

Negatif sayıda ürün üretilmeyeceği için değişkenler en düşük 0 değerini alabilirler. Bu durum $C14:E14 \geq 0$ şeklinde Çözücüde kısıt olarak belirtilmiştir. Alternatif olarak Çözücüde "Kısıtlanmamış Değişkenleri Pozitif Yap" seçeneği de

Uretim Kararlari

	LCD TV	Muzik Calar	Hoparlor
Uretim Miktari	199	200	2

Part Requirements by Product

Parca Adi	LCD TV	Muzik Calar	Hoparlor
Govde	1	1	0
LCDEkran	1	0	0
Hoparlor	2	2	1
Guc Unitesi	1	1	0
Elektronik Parca	2	1	1

Malzeme Stok Miktarlari

Kullanilan	Mevcut
399	450
199	250
800	800
399	450
600	600

Kısıtlar

Örnek model, kullanılacak malzeme miktarlarını sınırlayan 5 tane kısıt içermektedir. Çözücüde $Number_used \leq Number_available$ kısıtı bu durumu ifade etmektedir. Bu aynı zamanda $G18:G22 \leq H18:H22$ anlamına gelmektedir.

Amaç Fonksiyonu

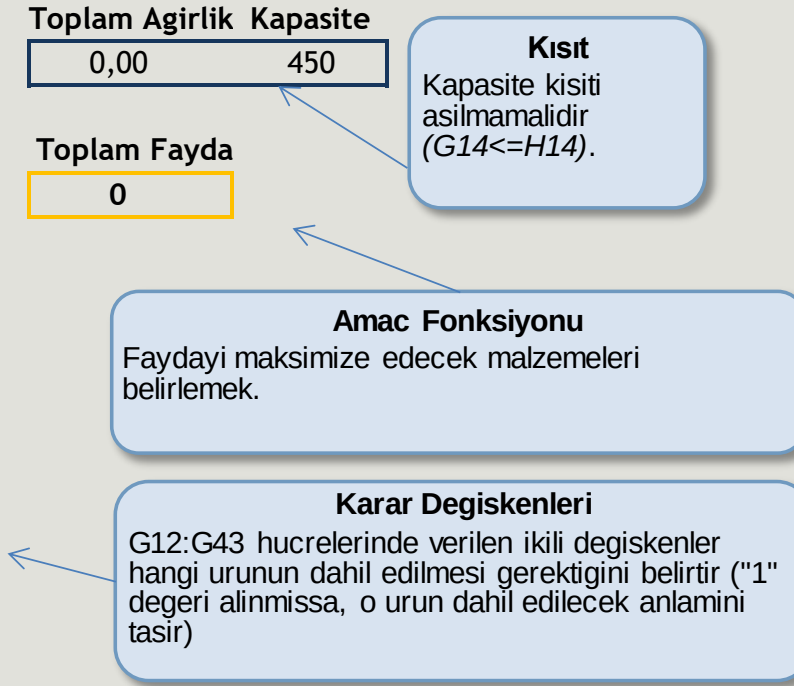
Modeldeki amacımız G24 hucresinde hesaplanan karı (Total_profit) maksimize etmektir.

Birim Kar	\$75	\$50	\$35	24.995,00
-----------	------	------	------	-----------

Excel Solver-Örnek 6: Sırt Çantası (Knapsack) Problemi

32 adet tıbbi malzemenin bir kargo aracına yüklenerek savaş bölgesine taşınması planlanmaktadır. Her bir malzeme, bir diğerine göre daha öncelikli olabileceği için tanımlanmış olan bir "fayda"ya sahiptir. Kargo aracının toplam kapasitesi 450 kg'dır ve malzemelerin ağırlıkları tabloda verilmiştir. Amac, faydayı maksimize edecek şekilde kapasite sınırını aşmadan hangi ürünlerin kargo aracına dahil edileceğine karar vermektir.

Malzeme	Ağırlık	Fayda	Dahil Et?
1	20,0	49	0
2	27,0	27	0
3	26,0	77	0
4	26,0	18	0
5	24,0	33	0
6	28,0	45	0
7	20,0	68	0
8	27,0	16	0
9	22,0	40	0
10	20,0	30	0
11	29,0	74	0
12	28,0	39	0
13	21,0	57	0
14	20,0	69	0
15	27,0	50	0
16	27,0	41	0
17	25,0	71	0
18	23,0	17	0
19	23,0	67	0
20	23,0	51	0
21	27,0	70	0
22	29,0	74	0
23	24,0	39	0
24	27,0	41	0
25	29,0	73	0
26	29,0	33	0
27	29,0	27	0
28	23,0	38	0
29	24,0	65	0
30	20,0	38	0
31	21,0	25	0
32	21,0	38	0



Excel Solver-Örnek 7

Bir işletme, aşağıdaki özellikleri taşıyan ürün üretmektedir:

$$\text{Yanma Noktası} \geq 2800$$

$$\text{Özgöl Ağırlığı} \geq 1.00$$

$$\alpha \text{ muhtevası} \geq \% 8 \text{ hacim olarak}$$

$$\beta \text{ muhtevası} \geq \% 3 \text{ hacim olarak}$$

Ürün, litresi sırası ile 16 TL, 20 TL ve 25 TL olan A, B ve C yarı mamullerinden elde edilebilmektedir. Bu üç yarı mamulün özellikleri aşağıdaki gibidir.

	A	B	C
Yanma noktası	2800	2700	2900
Özgöl ağırlığı	0.95	1.20	1.10
α muhtevası	7	8	11
β muhtevası	4	5	3

Üretici, yukarıdaki koşullara uygun olarak ürününü en düşük maliyetle üretmek istemektedir. Buna göre problemin doğrusal programlama modeli nasıl kurulur ve çözülür?

Excel Solver-Örnek 7

Çözüm:

Karar Değişkenleri

x_1 = Üründeki A oranı

x_2 = Üründeki B oranı

x_3 = Üründeki C oranı

Amaç fonksiyonu

$$\text{Min } Z = 16x_1 + 20x_2 + 25x_3$$

Problem Kısıtları

$$x_1 + x_2 + x_3 = 1$$

$$280x_1 + 270x_2 + 290x_3 \geq 280 \quad (\text{Yanma noktası kısıtı})$$

$$0.95x_1 + 1.20x_2 + 1.10x_3 \geq 1.00 \quad (\text{Özgül ağırlık kısıtı})$$

$$7x_1 + 8x_2 + 11x_3 \geq 8 \quad (\alpha \text{ muhtevası kısıtı})$$

$$4x_1 + 5x_2 + 3x_3 \geq 3 \quad (\beta \text{ muhtevası kısıtı})$$

EXCEL SOLVER ÇÖZÜMÜ

	A	B	C		X1	X2	X3	Z
Yanma noktası	2800,00	2700,00	2900,00		0,71	0,06	0,24	18,35
Özgül ağırlığı	0,95	1,20	1,10		Toplam	1,00		
α muhtevası(%)	7,00	8,00	11,00					
β muhtevası(%)	4,00	5,00	3,00					
Sınırlar				Yarımamül Fiyatları				
Yanma Noktası	2800,00	2817,65		A	16,00			
Özgül Ağırlık	1,00	1,00		B	20,00			
α muhtevası(%)	8,00	8,00		C	25,00			
β muhtevası(%)	3,00	3,82						

Excel Solver-Örnek 8

Bir kuruluş, büyük bir şehirde yeni mağazalar açmayı planlamaktadır. Açılacak mağazaların sayısı maksimum 11'dir.

Mağazalar üç farklı boyutta olabilir: Tip-1, Tip-2 ve Tip-3. Tip-1 mağazalar 24 saat boyunca açık olacağı için Tip-2 mağazalara göre daha fazla çalışan gerektirmektedir. Bu mağazaların yatırım maliyetleri ve çalışan sayılarına ilişkin bilgiler aşağıdaki tabloda verilmektedir.

	Tip-1	Tip-2	Tip-3
Gerekli Yatırım (*milyon PB)	4.125	8.25	12.375
Gerekli Çalışan Sayısı	30	15	45

Kuruluş, toplam yatırım bütçesi olarak 82.5 milyon PB ayırmıştır ve en fazla 300 çalışan istihdam etmeyi planlamaktadır.

Tip-1, Tip-2 ve Tip-3 mağazaların sırasıyla 1.2 milyon PB, 2 milyon PB ve 2.6 milyon PB yıllık getirisi olacağı düşünülmektedir.

Kuruluş, karını maksimize etmek için hangi tip mağazadan kaç tane açmalıdır?

	A	B	C	D	E	F	G	H	I
1									
2									
3		Tip-1	Tip-2	Tip-3				Gerçekleşen	Kısıt
4	Gerekli Yatırım (*milyon PB)	4,125	8,25	12,375			Toplam Yatırım:	0	82,5
5	Gerekli Çalışan Sayısı	30	15	45			Toplam Çalışan:	0	300
6	Açılması Durumunda Getiri	1,2	2	2,6			Açılan Mağaza:	0	11
7									
8		X1	X2	X3			Toplam Getiri:	0	
9	Açılacak Mağaza Sayısı	0	0	0					
10	Açılan Mağazaların Yatırımı	0	0	0					
11	Açılan Mağazalardaki Çalışan	0	0	0					
12	Açılan Mağazaların Getirisi	0	0	0					
13									

Excel Solver-Örnek 8

Çözüm

	A	B	C	D
1				
2				
3		Tip-1	Tip-2	Tip-3
4	Gerekli Yatırım (*milyon PB)	4,125	8,25	12,375
5	Gerekli Çalışan Sayısı	30	15	45
6	Açılması Durumunda Getiri	1,2	2	2,6
7				
8		X1	X2	X3
9	Açılacak Mağaza Sayısı	0	0	0
10	Açılan Mağazaların Yatırımı	0	0	0
11	Açılan Mağazalardaki Çalışan	0	0	0
12	Açılan Mağazaların Getirisi	0	0	0

Solver Parameters

Set Objective:

\$H\$8

To:

☒ Max

☐ Min

☐ Value Of:

0

By Changing Variable Cells:

\$B\$9:\$D\$9

Subject to the Constraints:

\$B\$9:\$D\$9 = integer

\$H\$4 <= \$I\$4

\$H\$5 <= \$I\$5

\$H\$6 <= \$I\$6

Add

Change

Delete

Reset All

Load/Save

☒ Make Unconstrained Variables Non-Negative

Select a Solving Method:

Simplex LP

Options

Solving Method

Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.

Help

Solve

Close

	A	B	C	D	E	F	G	H	I
1									
2									
3		Tip-1	Tip-2	Tip-3				Gerçekleşen	Kısıt
4	Gerekli Yatırım (*milyon PB)	4,125	8,25	12,375			Toplam Yatırım:	82,5	82,5
5	Gerekli Çalışan Sayısı	30	15	45			Toplam Çalışan:	195	300
6	Açılması Durumunda Getiri	1,2	2	2,6			Açılan Mağaza:	11	11
7									
8		X1	X2	X3			Toplam Getiri:	20,4	
9	Açılacak Mağaza Sayısı	2	9	0					
10	Açılan Mağazaların Yatırımı	8,25	74,25	0					
11	Açılan Mağazalardaki Çalışan	60	135	0					
12	Açılan Mağazaların Getirisi	2,4	18	0					
13									

Excel Solver-Örnek 9

Makine atama problemi – Amaç: Toplam maliyetin minimizasyonu

Tabloda, operatörlerin makinelere atanması sonucu oluşacak maliyetler verilmiştir.

Her makineye bir operatör atanmak koşuluyla, toplam maliyeti minimize etmek için hangi operatörün hangi makineye atanması gerektiğini bulunuz.

	T1	T2	T3	T4	T5
A	20	15	25	25	29
B	13	19	30	13	19
C	20	17	14	12	15
D	14	20	20	16	24
E	14	16	19	11	22

Excel Solver-Örnek 9

Çözüm

	A	B	C	D	E	F	G	H
1								
2		T1	T2	T3	T4	T5		
3	A	20	15	25	25	29		
4	B	13	19	30	13	19		
5	C	20	17	14	12	15		
6	D	14	20	20	16	24		
7	E	14	16	19	11	22		
8								
9								
10		T1	T2	T3	T4	T5		Atanan
11	A	0	0	0	0	0		0
12	B	0	0	0	0	0		0
13	C	0	0	0	0	0		0
14	D	0	0	0	0	0		0
15	E	0	0	0	0	0		0
16								
17	Atanan	0	0	0	0	0		
18								
19							Maliyet:	0
20								

Solver Parameters

Set Objective:

\$H\$19

To: ☐ Max ☒ Min ☐ Value Of:

0

By Changing Variable Cells:

\$B\$11:\$F\$15

Subject to the Constraints:

\$B\$17:\$F\$17 = 1
\$H\$11:\$H\$15 = 1

Add

Change

Delete

Reset All

Load/Save

☒ Make Unconstrained Variables Non-Negative

Select a Solving Method:

Simplex LP

Options

Solving Method

Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.

Help

Solve

Close

H19										
	A	B	C	D	E	F	G	H	I	J
1										
2		T1	T2	T3	T4	T5				
3	A	20	15	25	25	29				
4	B	13	19	30	13	19				
5	C	20	17	14	12	15				
6	D	14	20	20	16	24				
7	E	14	16	19	11	22				
8										
9										
10		T1	T2	T3	T4	T5		Atanan		
11	A	0	1	0	0	0		1		
12	B	0	0	0	0	1		1		
13	C	0	0	1	0	0		1		
14	D	1	0	0	0	0		1		
15	E	0	0	0	1	0		1		
16										
17	Atanan	1	1	1	1	1				
18										
19							Maliyet:	73		
20										

Çözüm

Excel Solver-Ödev 1 TEMİZLİK MALZEMESİ

ABC şirketi yeni bir yer silme sıvısı üretecektir. Bu sıvının içinde A temizleyici maddesinden en az %20 en fazla %40, B temizleyici maddesinden en az %25 en fazla %50 ve C temizleyici maddesinden en az %35 en fazla %50 olacaktır. Şirket, ürününü üretmek için 3 ayrı hammadde kullanmaktadır. Bu hammaddelerin içerikleri aşağıdaki tabloda verilmiştir.

Hammadde	A maddesi	B maddesi	C maddesi
I	%60	%30	%10
II	%25	%25	%50
III	%20	%50	%30

Hammadde I, II ve III' ün kg maliyetleri sırasıyla 1.5 TL, 1.55 TL, 1.6 TL'dir. Şirket en az maliyetle talebi karşılayacak yer silme sıvısının üretimi için doğrusal programlama modeli oluşturup çözmek istemektedir.

Geliştirilen matematiksel modele göre Excel Solver kullanarak çözümü bulunuz.

Excel Solver-Ödev 1

Karar değişkenleri

Karışım probleminde, her biri ürün içinde yer alacak hammadde miktarına karşılık gelen 3 karar değişkeni tanımlanmalıdır. Bunlar;

P_i = ürünün içinde yer alacak i hammaddesi oranı

Amaç fonksiyonu

Problemin amacı kullanılan hammadde maliyetini minimize etmektir. Dolayısıyla, karar değişkenleri ile o karar değişkenine karşılık gelen maliyet değerleri çarpılıp sonra da tüm değerler toplanarak amaç fonksiyonu elde edilir.

$$\text{Min } Z = 1.5 P_1 + 1.55P_2 + 1.6P_3$$

Excel Solver-Ödev 1

Problemnin kısıtları

Problemdeki ilk üç kısıt, ürün içinde A, B ve C maddelerinden bulunması gereken en az miktarı sağlayan kısıtlardır.

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \geq 0.20$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \geq 0.25$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \geq 0.35$$

Problemdeki ikinci grup üç kısıt da, ürünün içinde A, B ve C maddelerinden bulunması gereken en fazla miktarı sağlayan kısıtlardır.

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \leq 40$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \leq 50$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \leq 50$$

Excel Solver-Ödev 1

Üç hammaddeden kullanılan oranların toplamının %100 olmasını sağlayan kısıt da aşağıdaki şekilde yazılmalıdır.

$$P_1 + P_2 + P_3 = 1$$

Son olarak karar değişkenlerinin negatif olmama kısıtları da aşağıda gösterildiği gibi modele eklenmelidir.

$$P_i \geq 0$$

Excel Solver-Ödev 1

Amaç fonksiyonu ve kısıtlar bir araya getirilerek karışım modeli aşağıdaki gibi oluşturulur.

$$\text{Min } Z = 1.5 P_1 + 1.55P_2 + 1.6P_3$$

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \geq 0.20$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \geq 0.25$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \geq 0.35$$

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \leq 40$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \leq 50$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \leq 50$$

$$P_1 + P_2 + P_3 = 1$$

$$P_i \geq 0$$

Excel Solver-Ödev 2

Bir rafineri 2 tür kurşunsuz benzin üretimi yapmaktadır. 1. tipin varil fiyatı 48\$, 2. tipin varil fiyatı ise 53\$'dır. Her iki tip benzin de Tablo 1'de verilen özellikleri karşılamak zorundadır. Karışımında kullanılan bileşenler ve özellikleri de Tablo 2'de verilmiştir. Haftalık karı maksimize etmek için iki tip benzine hangi bileşenlerden hangi miktarlarda karıştırılmalıdır? Geliştirilen matematiksel modele göre Excel Solver kullanarak çözümü bulunuz.

Tablo 1

Benzin tipi	Minimum oktan oranı	Maksimum talep (varil/hafta)	Minimum dağıtım (varil/hafta)
1. Tip Benzin	87	80.000	60.000
2. Tip Benzin	93	40.000	15.000

Tablo 2

Benzin Bileşenleri	Oktan oranı	Arz miktarı (varil)	Maliyet (\$/varil)
1	86	70.000	33
2	96	60.000	37

Excel Solver-Ödev 2

Karar değişkenleri

Bu problemdeki kontrol değişkenleri, benzinlere karıştırılması gereken iki bileşenin miktarlarıdır.

x_{ij} = j tip benzine karıştırılan i bileşeninin varil cinsinden haftalık miktarı

$$i = 1, 2 ; \quad j = 1, 2$$

Amaç Fonksiyonu

Benzin bileşenlerinin doğrusal olarak karıştığı kabul edilirse, 1. tip benzin miktarı $x_{11} + x_{21}$, ikinci tip benzin miktarı ise $x_{12} + x_{22}$ olarak verilir. Benzer şekilde kullanılan 1. ve 2. bileşenlerin toplam miktarları da sırasıyla $x_{11} + x_{12}$ ve $x_{21} + x_{22}$ olarak verilebilir. Amaç fonksiyonu şu şekilde formüle edilebilir:

$$Kar = \text{satış} - \text{maliyet}$$

$$Max Z = 48(x_{11} + x_{21}) + 53(x_{12} + x_{22}) - 33(x_{11} + x_{12}) - 37(x_{21} + x_{22}) \quad \text{haftalık kar}$$

$$Max Z = 15x_{11} + 20x_{12} + 11x_{21} + 16x_{22}$$

Excel Solver-Ödev 2

Problemin Kısıtları

Minimum oktan oranı ihtiyacını karşılamak için, iki bileşenin değişik miktarlarda karıştırılması durumunda ortaya çıkacak oktan seviyesini belirlemeliyiz. Bileşenlerin yine doğrusal olarak karıştığını kabul ediyoruz. Bileşenlerin karışımındaki oktan seviyesi ağırlıklı ortalama alınarak hesaplanabilir. Karışımındaki toplam oktanı, varil miktarına bölerek buluruz.

Böylece, 1. tip benzin için minimum oktan oranı kısıtı

$$\frac{86x_{11} + 96x_{21}}{x_{11} + x_{21}} \geq 87$$

Bu kısıt lineer değildir fakat eşitsizliğin her iki tarafını

$x_{11} + x_{21}$ ile çarparsak;

$$-x_{11} + 9x_{21} \geq 0$$

Aynı şekilde, 2. tip benzin için minimum oktan oranı kısıtı

$$\frac{86x_{12} + 96x_{22}}{x_{12} + x_{22}} \geq 93$$

Bu kısıt lineer değildir fakat eşitsizliğin her iki tarafını

$x_{12} + x_{22}$ ile çarparsak;

$$-7x_{12} + 3x_{22} \geq 0$$

Excel Solver-Ödev 2

Benzin tipleri için minimum ve maksimum dağıtım gereksinimi kısıtları ise şu şekilde formüle edilebilir:

$$x_{11} + x_{21} \geq 60.000 \quad (1. \text{tip benzin için minimum dağıtım miktarı})$$

$$x_{12} + x_{22} \geq 15.000 \quad (2. \text{tip benzin için minimum dağıtım miktarı})$$

$$x_{11} + x_{21} \leq 80.000 \quad (1. \text{tip benzin için maksimum talep miktarı})$$

$$x_{12} + x_{22} \leq 40.000 \quad (2. \text{tip benzin için maksimum talep miktarı})$$

Benzer şekilde, bileşenlerin arz kısıtları:

$$x_{11} + x_{12} \leq 70.000 \quad (1. \text{bileşenin arz miktarı})$$

$$x_{21} + x_{22} \leq 60.000 \quad (2. \text{bileşenin arz miktarı})$$

Negatif olmama kısıtları:

$$x_{11}, x_{12}, x_{21}, x_{22} \geq 0$$

Excel Solver-Ödev 2

Böylece lineer programlama modelinin tamamı aşağıdaki şekilde yazılabilir.

$$\text{Max } Z = 15x_{11} + 20x_{12} + 11x_{21} + 16x_{22}$$

$$-x_{11} + 9x_{21} \geq 0$$

$$-7x_{12} + 3x_{22} \geq 0$$

$$x_{11} + x_{21} \geq 60.000$$

$$x_{12} + x_{22} \geq 15.000$$

$$x_{11} + x_{21} \leq 80.000$$

$$x_{12} + x_{22} \leq 40.000$$

$$x_{11} + x_{12} \leq 70.000$$

$$x_{21} + x_{22} \leq 60.000$$

$$x_{11}, x_{12}, x_{21}, x_{22} \geq 0$$

Excel Solver-Ödev 2

Doğrusallık varsayımları altında, modelimiz karışım gereksinimlerini tam olarak sağlamaktadır. Fakat satışlardan elde edilen gelir ve maliyetler, amaç fonksiyonunda görünmemektedir.

1. ve 2. tip benzin miktarlarına sırasıyla g_1 ve g_2 dersek;

$$g_1 = x_{11} + x_{21} \text{ veya}$$

$$x_{11} + x_{21} - g_1 = 0 \text{ (1. tip benzinin varil cinsinden toplam miktarı)}$$

Aynı şekilde;

$$g_2 = x_{12} + x_{22} \text{ veya}$$

$$x_{12} + x_{22} - g_2 = 0 \text{ (2. tip benzinin varil cinsinden toplam miktarı)}$$

1. ve 2. bileşenlere ise sırasıyla h_1 ve h_2 dersek;

$$x_{11} + x_{12} - h_1 = 0 \text{ (1. tip bileşenin varil cinsinden toplam miktarı)}$$

$$x_{21} + x_{22} - h_2 = 0 \text{ (2. tip bileşenin varil cinsinden toplam miktarı)}$$

Excel Solver-Ödev 2

Bu yeni ilişkilerden yararlanarak önceki modeli daha basit bir şekilde ifade edebiliriz.

Amaç Fonksiyonu:

$$\text{Maksimum kar } Z = 48g_1 + 53g_2 - 33h_1 - 37h_2$$

Kısıtlar:

$$x_{11} + x_{21} - g_1 = 0 \qquad g_1 \geq 60.000$$

$$x_{12} + x_{22} - g_2 = 0 \qquad g_2 \geq 15.000$$

$$x_{11} + x_{12} - h_1 = 0 \qquad g_1 \leq 80.000$$

$$x_{21} + x_{22} - h_2 = 0 \qquad g_2 \leq 40.000$$

$$-x_{11} + 9x_{21} \geq 0 \qquad h_1 \leq 70.000$$

$$-7x_{12} + 3x_{22} \geq 0 \qquad h_2 \leq 60.000$$

$$x_{11}, x_{12}, x_{21}, x_{22}, g_1, g_2, h_1, h_2 \geq 0$$

	A	B	C	D	E	F	G	H	I	J
1										
2	Benzin Tipi	Minimum Oktan Oranı	Maksimum Talep (varil/hafta)	Maksimum Dağıtım (varil/hafta)	Satış Fiyatı		Benzin Bileşenleri	Oktan Oranı	Arz Miktarı (varil)	Maliyet (\$ / varil)
3	1.Tip Benzin	87	80000	60000	48		B1	86	70000	33
4	2.Tip Benzin	93	40000	15000	53		B2	96	60000	37
5										
6										
7	X11	X21	X12	X22						
8	58000	22000	12000	28000			Toplam Kar	1800000		
9										
10										
11	Üretim Miktarı		Kullanım Miktarı							
12	1. Tip Benzin	80000	B1	70000						
13	2. Tip Benzin	40000	B2	50000						
14										
15	Toplam Oktan		ÖNEMLİ NOT! Oktan Kısıtı: $B_{19} \geq B_6 * B_{15}$ $B_{20} \geq B_7 * B_{16}$ Bu yol, bölme yapmayarak paydanın sıfır olmasını engelliyor.							
16	1. Tip Benzin	7100000								
17	2. Tip Benzin	3720000								
18										
19										
20										
21										

Xij
i: bileşen
j: benzin tipi

Excel Solver-Ödev 3

Example 1

Diet Problem: Set-Up (1 of 7)

Problem Statement

- Consider the problem of diet optimization based on cost and different nutritional factors
- There are four different types of food: Brownies, Ice Cream, Cola, and Cheese Cake, with nutrition values and cost per unit as follows:

	Brownies	Ice Cream	Cola	Cheese Cake
Calories	400	200	150	500
Chocolate	3	2	0	0
Sugar	2	2	4	4
Fat	2	4	1	5
Cost	\$0.50	\$0.20	\$0.30	\$0.80

Task:

- Find a minimum-cost diet that contains
 - at least 500 calories
 - at least 6 grams of chocolate
 - at least 10 grams of sugar
 - at least 8 grams of fat.

Example 1

Diet Problem: Set-Up (2 of 7)

- First, we must format our spreadsheet correctly to be entered into SOLVER
- Identify the decision variables (changing cells)
 - To begin we enter heading for each type of food in B2:E2
 - In the range B3:E3, we input random trial values for the amount of each food eaten (any values will work, but at least one should be positive)
 - In the example shown below, we indicate that we are considering eating 3 brownies, 0 scoops of chocolate ice cream, 1 bottle of cola, and 7 pieces of pineapple cheesecake:

	A	B	C	D	E
1	DECISION VARIABLES				
2		Brownies	Ice Cream	Cola	Cheese Cake
3	Eaten	3	0	1	7

Diet Problem: Set-Up (3 of 7)

- Write and enter objective function (target cell)
 - To see if the diet is optimal, we must determine its cost as well as the calories, chocolate, sugar, and fat it provides
 - In the range B7:E7 we reference the number of units, and in B8:E8 we input the per-unit cost for each available food
 - We compute the cost of the diet in cell B10 with the formula

$$= B7*B8 + C7*C8 + D7*D8 + E7*E8$$
 ...But it is usually easier to enter the formula

$$= \text{SUMPRODUCT} (B7:E7, B8:E8)$$
 ...And this is much easier to understand for anyone reading the spreadsheet
- The **=SUMPRODUCT** function requires two ranges as inputs
 - The first cell in range 1 is multiplied by the first cell in range 2, then the second cell in range 1 is multiplied by the second cell in range 2, and so on
 - All of these products are then added
 - Thus, in cell B10 the “=SUMPRODUCT” function computes total cost as

$$3*50 + 0*20 + 1*30 + 7*80 = 740 \text{ cents.}$$

Diet Problem: Set-Up (4 of 7)

- Now, the spreadsheet should look like:

	A	B	C	D	E
1	DECISION VARIABLES				
2		Brownies	Ice Cream	Cola	Cheese Cake
3	Eaten	3	0	1	7
4					
5	OBJECTIVE FUNCTION				
6		Brownies	Ice Cream	Cola	Cheese Cake
7	Eaten	=B3	=C3	=D3	=E3
8	Cost	50	20	30	80
9					
10	Total	740	= SUMPRODUCT (B7:E7, B8:E8)		

Diet Problem: Set-Up (5 of 7)

- Finally, we must set up the given problem constraints (for calories, chocolate, sugar, and fat)
 - To begin, we recreate the table in Excel that defines how many calories and units of chocolate, sugar, and fat are in each type of dessert
 - We can use this information to calculate total amounts based on the quantities of different decision variables
 - Next, take the =SUMPRODUCT of the number of items with the calories in each to calculate total calories in our dessert selection
= **SUMPRODUCT (B7:E7, B14:E14)**
 - Finally, indicate the limitations highlighted in the problem
 - Add a >= or <= to identify maximum versus minimum constraints in Column G, and use Column H to indicate those limits:

Example 1

Diet Problem: Set-Up (6 of 7)

- The formulas will look like:

	A	B	C	D	E	F	G	H
13		Brownies	Ice Cream	Cola	Cheese Cake	Totals		Required
14	Calories	400	200	150	500	=SUMPRODUCT(\$B\$7:\$E\$7,B14:E14)	>=	500
15	Chocolate	3	2	0	0	=SUMPRODUCT(\$B\$7:\$E\$7,B15:E15)	>=	6
16	Sugar	2	2	4	4	=SUMPRODUCT(\$B\$7:\$E\$7,B16:E16)	>=	10
17	Fat	2	4	1	5	=SUMPRODUCT(\$B\$7:\$E\$7,B17:E17)	>=	8

- The constraint values that will show up on your screen look like:

	A	B	C	D	E	F	G	H
13		Brownies	Ice Cream	Cola	Cheese Cake	Totals		Required
14	Calories	400	200	150	500	4850	>=	500
15	Chocolate	3	2	0	0	9	>=	6
16	Sugar	2	2	4	4	38	>=	10
17	Fat	2	4	1	5	42	>=	8

Example 1

Diet Problem: Set-Up (7 of 7)

- The complete LP to be entered into SOLVER now looks like:

	A	B	C	D	E	F	G	H
1	DECISION VARIABLES							
2		Brownies	Ice Cream	Cola	Cheese Cake			
3	Eaten	3	0	1	7			
4								
5	OBJECTIVE FUNCTION							
6		Brownies	Ice Cream	Cola	Cheese Cake			
7	Eaten	3	0	1	7			
8	Cost	50	20	30	80			
9								
10	Total	740						
11								
12	CONSTRAINTS							
13		Brownies	Ice Cream	Cola	Cheese Cake	Totals		Required
14	Calories	400	200	150	500	4850	>=	500
15	Chocolate	3	2	0	0	9	>=	6
16	Sugar	2	2	4	4	38	>=	10
17	Fat	2	4	1	5	42	>=	8

GAMS Kurulumu ve Temel Özellikleri

GAMS ile Modellemeye Giriş

3

GAMS Programının İndirilmesi

NOT: İndirmek için
size verilen linki
kullanınız!



Download GAMS Distribution 25.0.2 (January, 31, 2018)

Please consult the [release notes](#) before downloading a system. Here are the [detailed platform descriptions](#) and [installation notes](#). The GAMS distribution includes the [documentation](#) in electronic form.

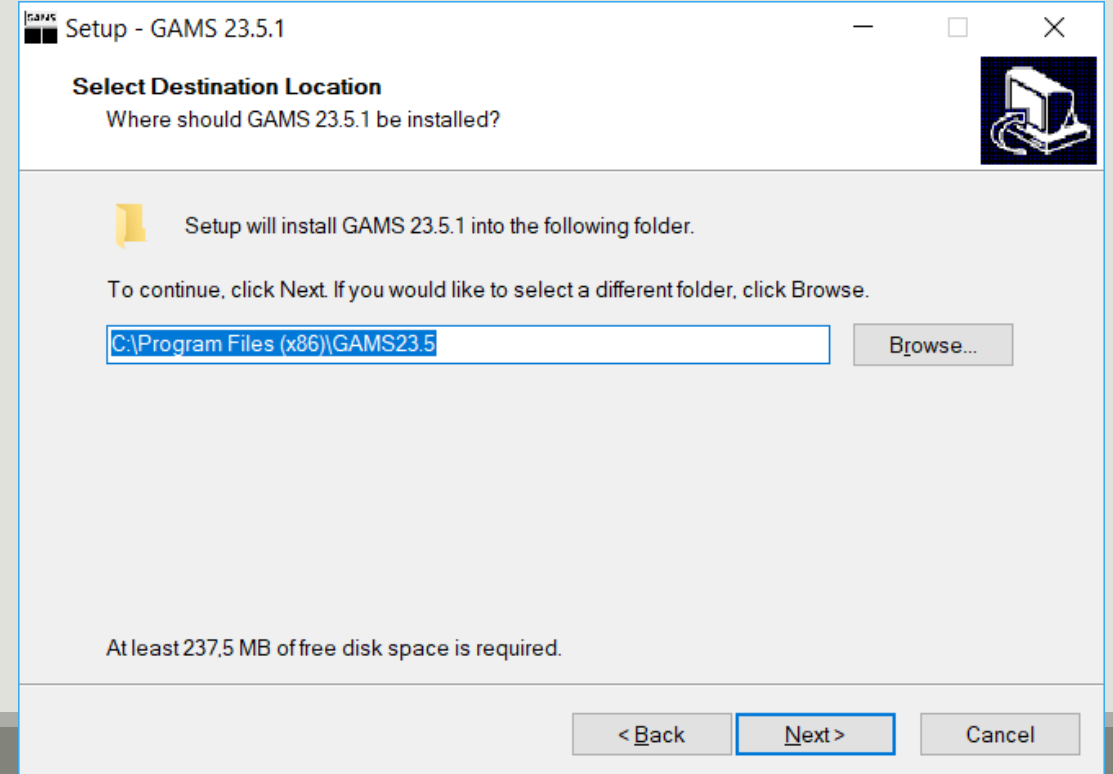
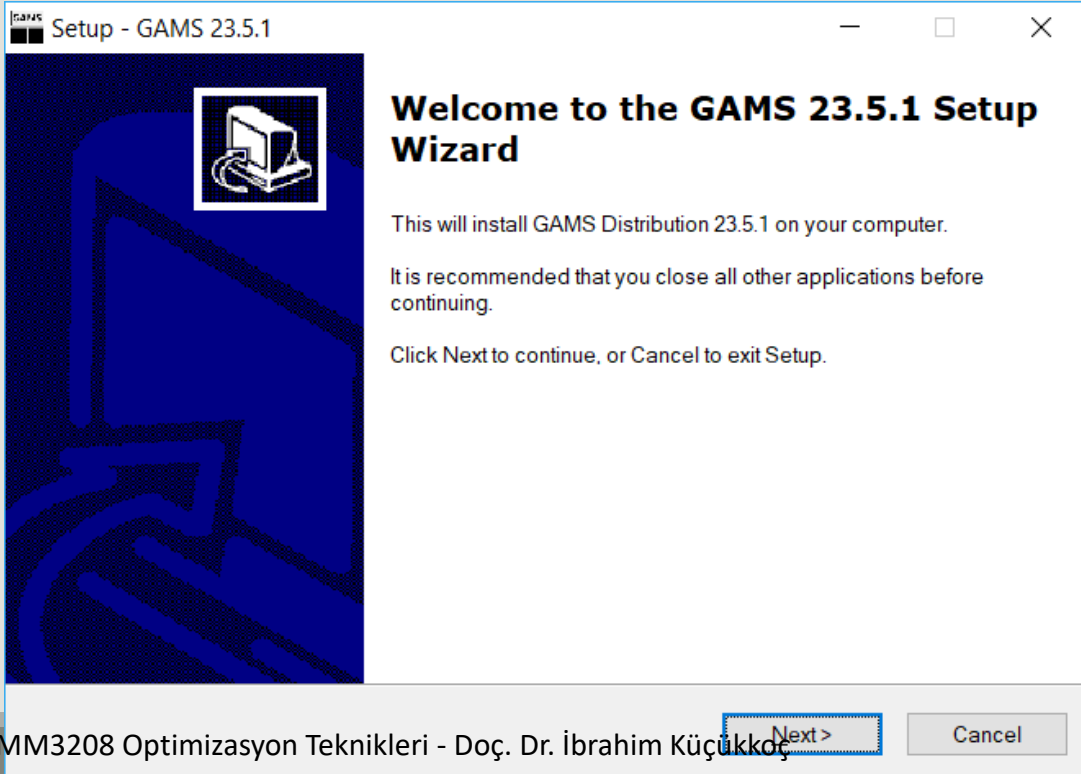
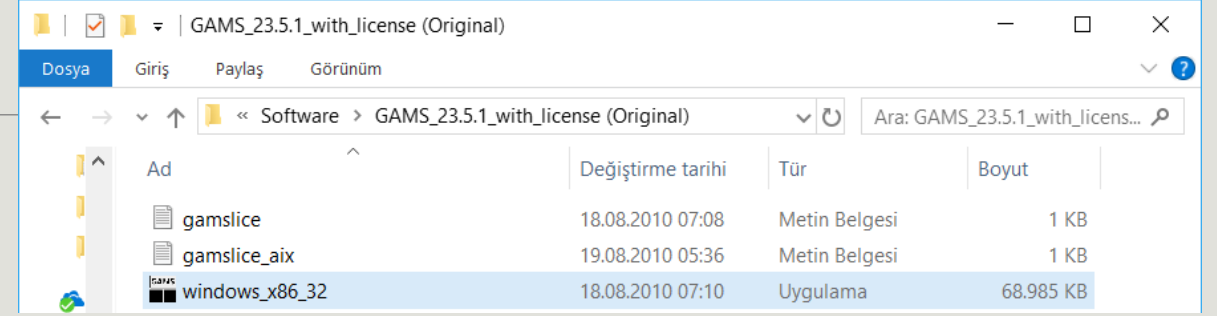
To deliver GAMS with the best performance we are using the [Amazon CloudFront](#) web service, a global network of edge locations for content delivery.

Platform		
MS Windows 32 bit	Windows Vista or newer on AMD- or Intel-based (x86_32) architectures.	DOWNLOAD
MS Windows 64 bit	Windows Vista or newer on AMD- or Intel-based (x86_64) architectures.	DOWNLOAD

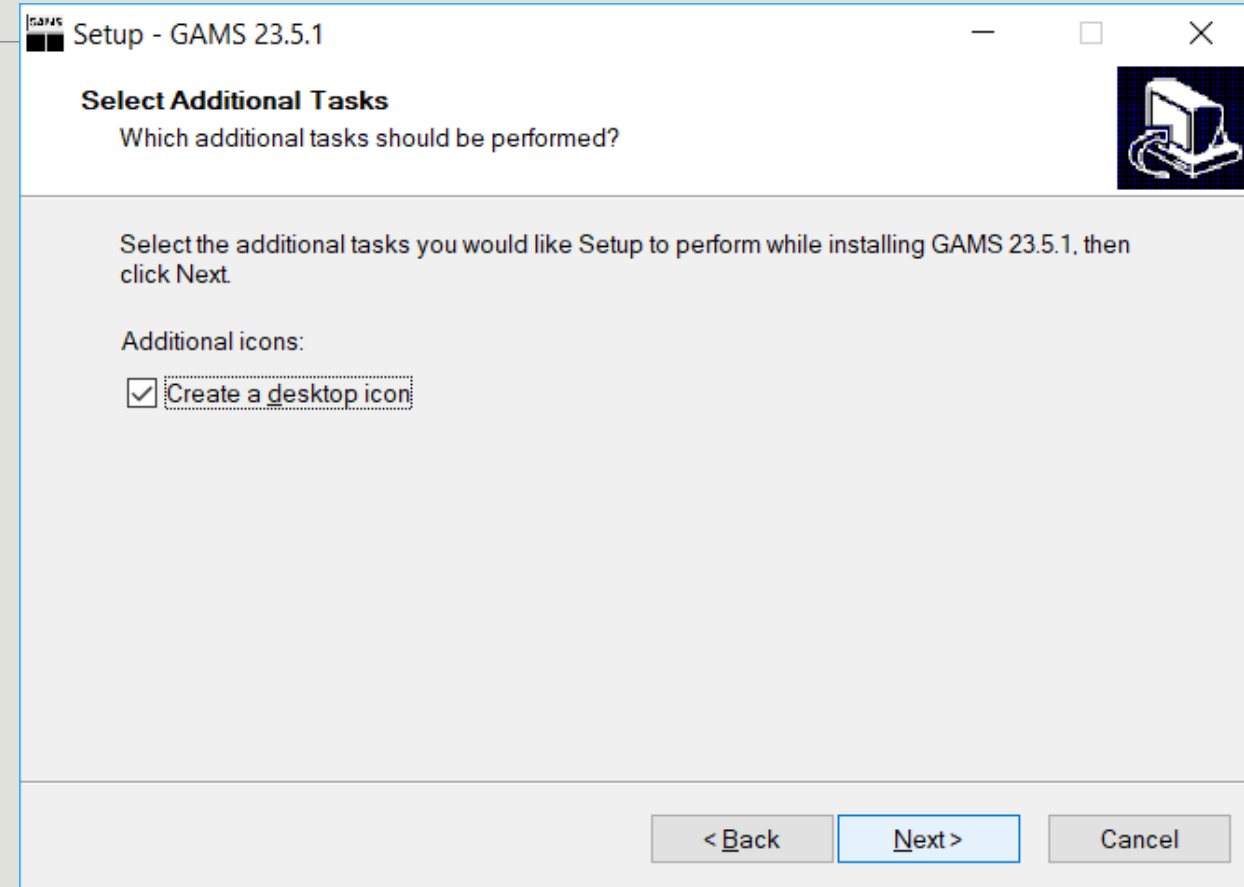
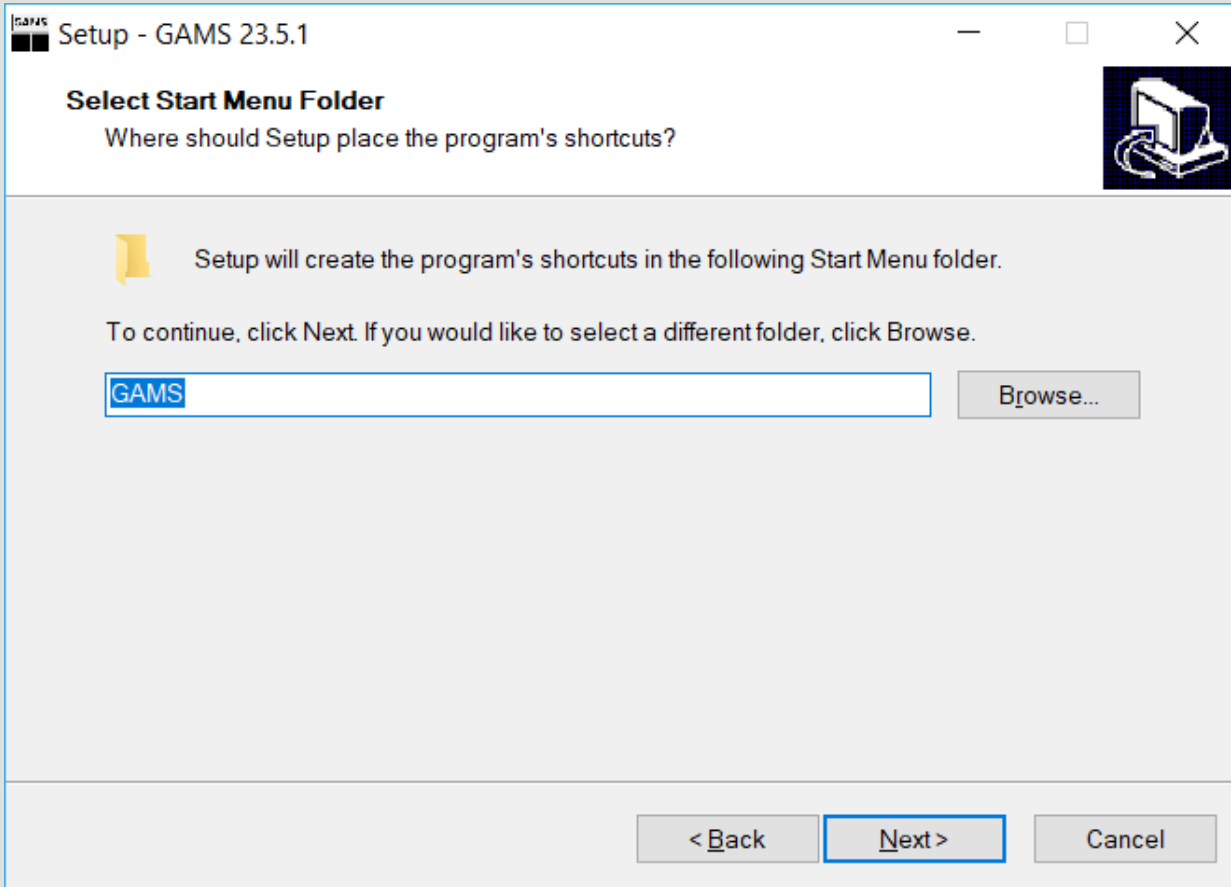
Yukarıda görüldüğü gibi programın **deneme sürümünü** ücretsiz olarak indirerek kullanabilirsiniz. Ayrıca satın alma olanakları da mevcuttur.

GAMS Kurulumu

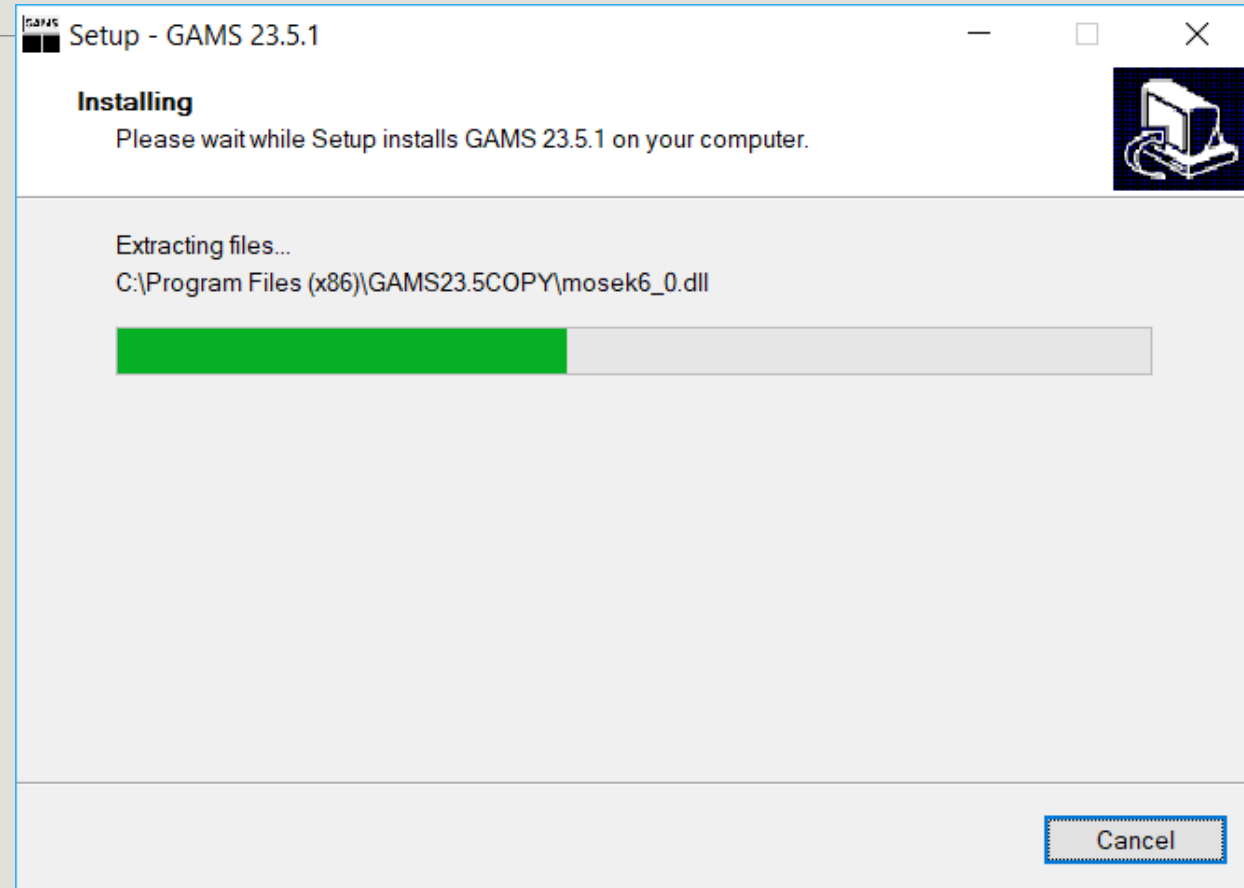
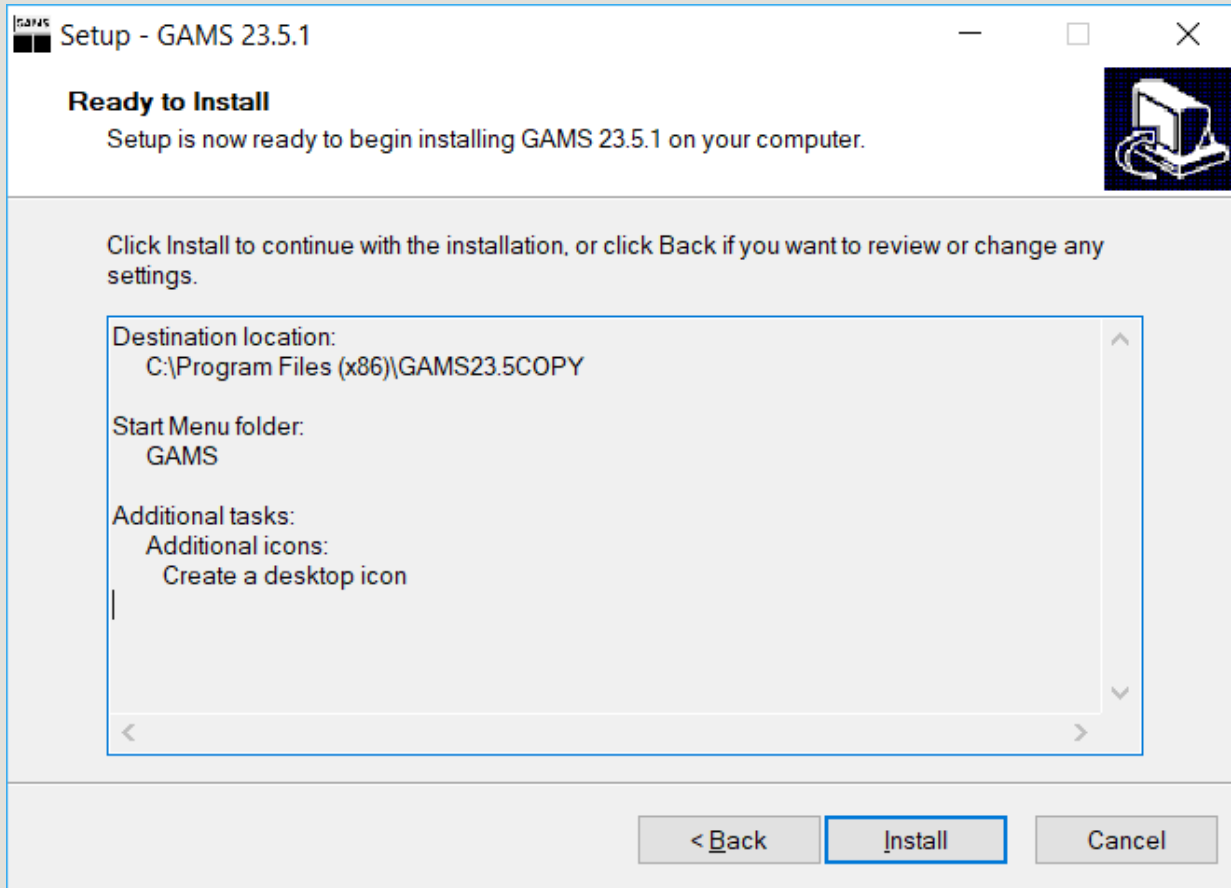
İndirilen kurulum dosyası tıkla-çalıştır yapılarak basitçe kurulur.



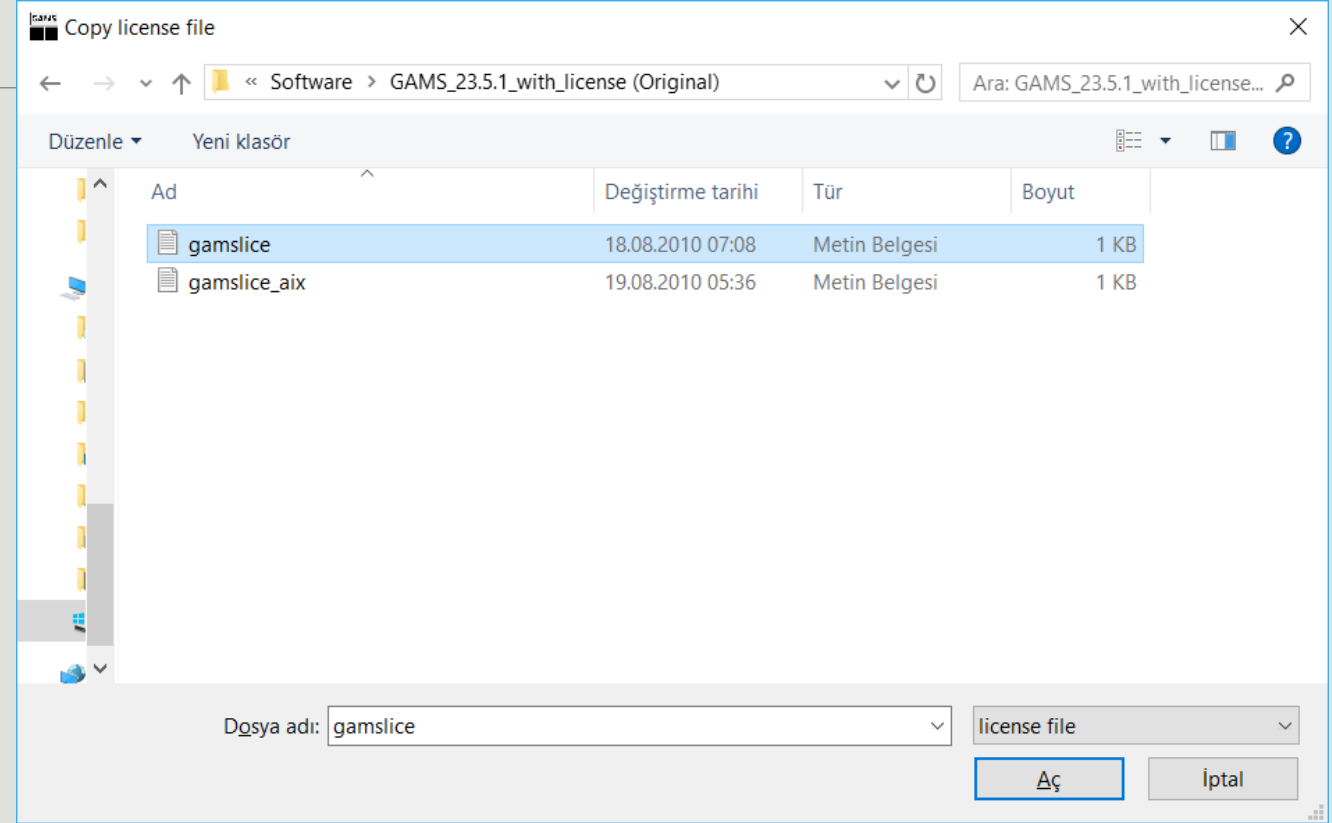
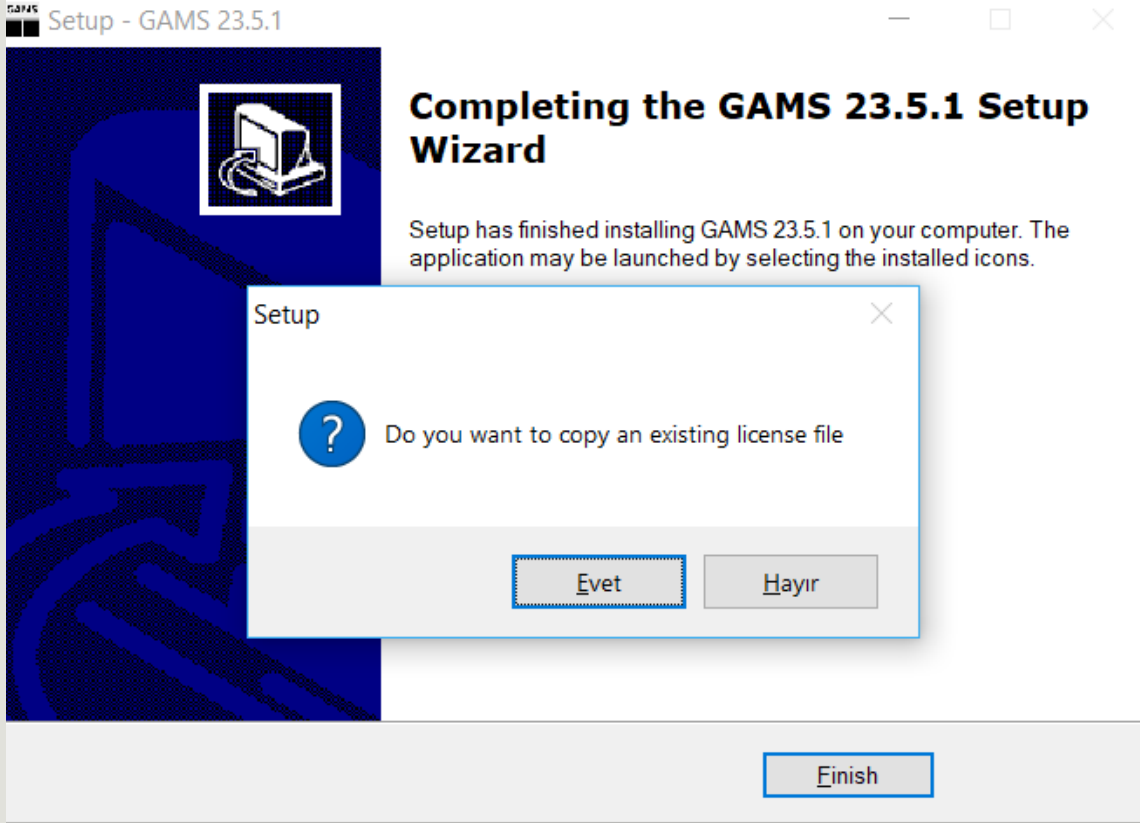
GAMS Kurulumu



GAMS Kurulumu

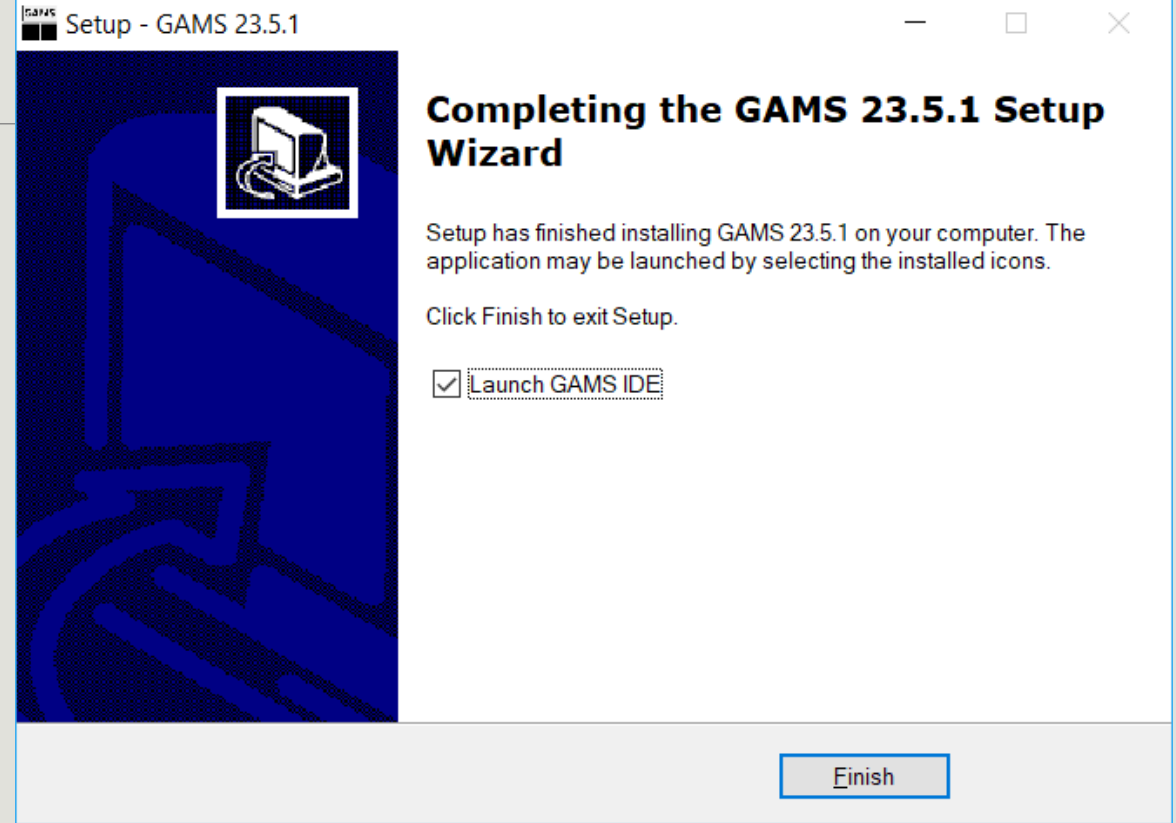
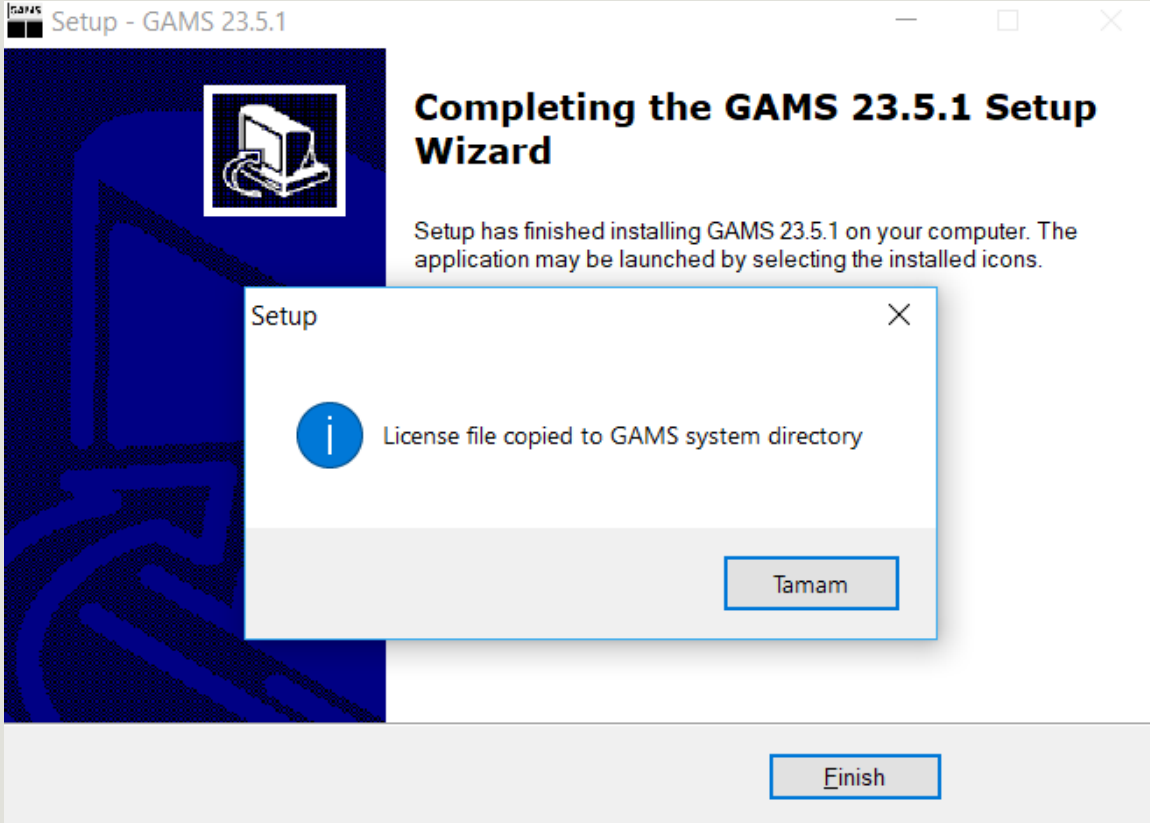


GAMS Kurulumu



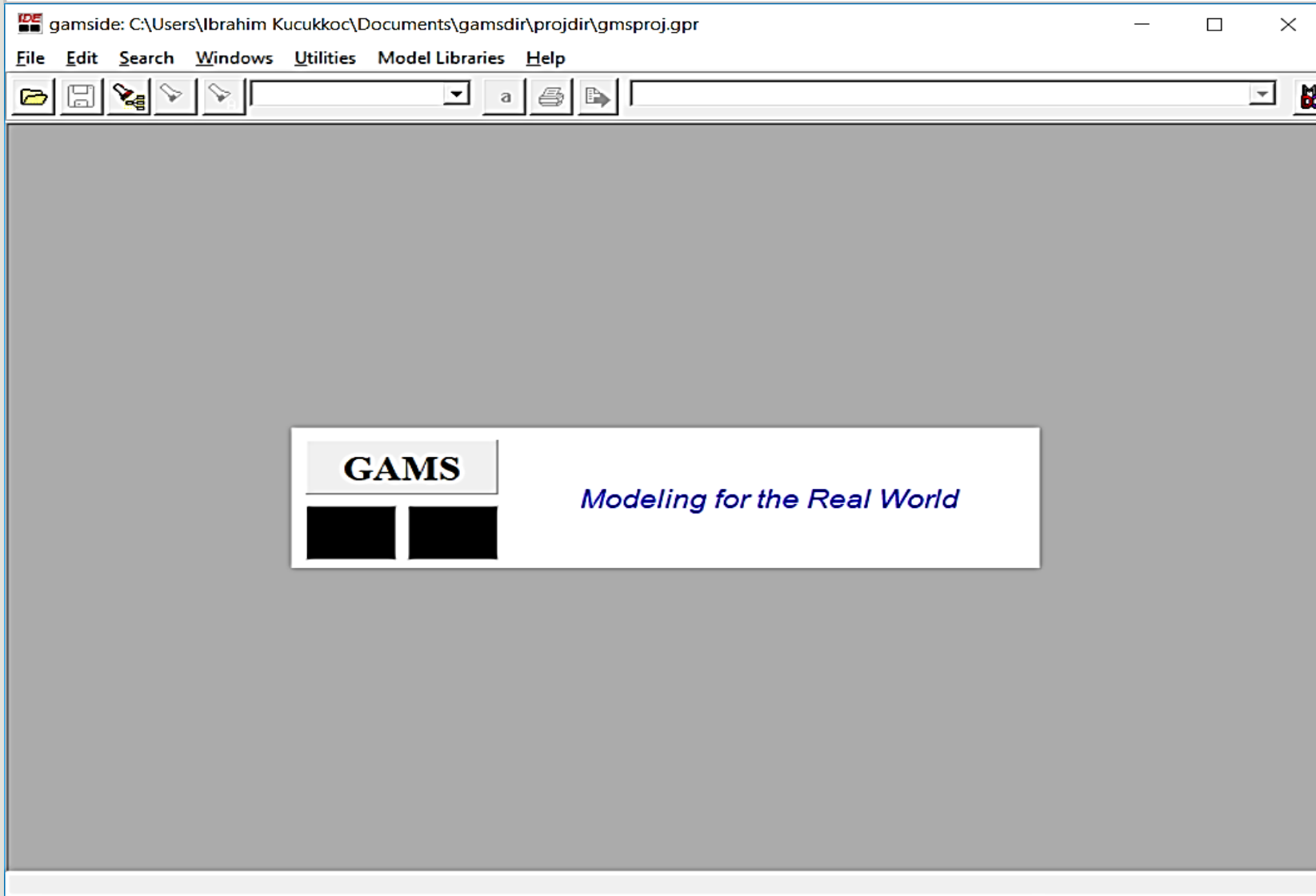
Şayet bir lisans dosyanız var ise, kurulum tamamlandıktan sonra lisans dosyasının yolu gösterilerek programın lisansı yapılmış olur.

GAMS Kurulumu



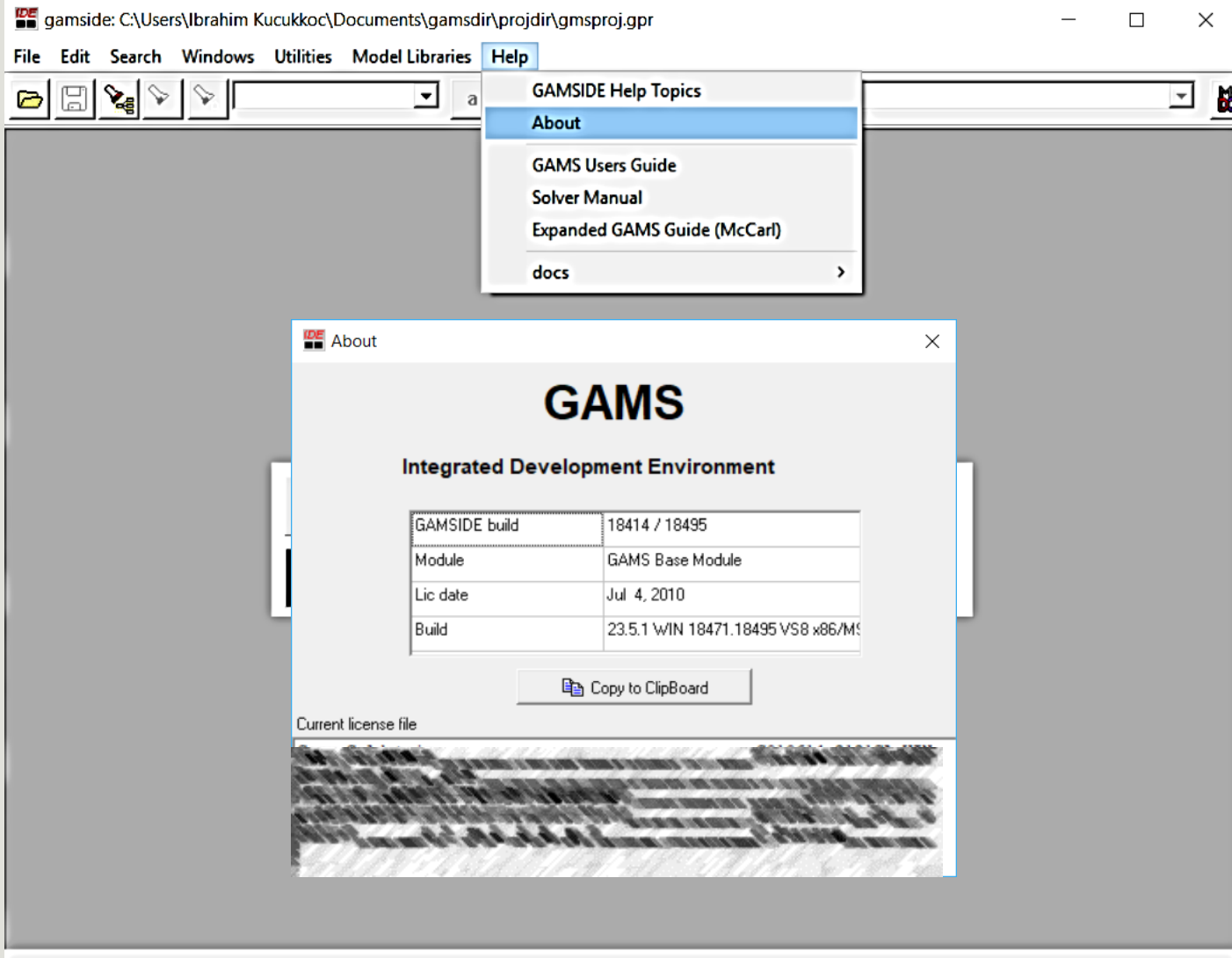
Burada gösterilen “Licence file copied to GAMS directory” mesajı, lisans dosyanızın başarılı bir şekilde yerleştirildiğini ifade etmektedir. Finish’e tıklayarak kurulumu bitirebilirsiniz.

GAMS Kurulumu



Artık GAMS, ilk modeli oluşturmak üzere kullanıma hazır hale gelmiştir.

GAMS Kurulumu

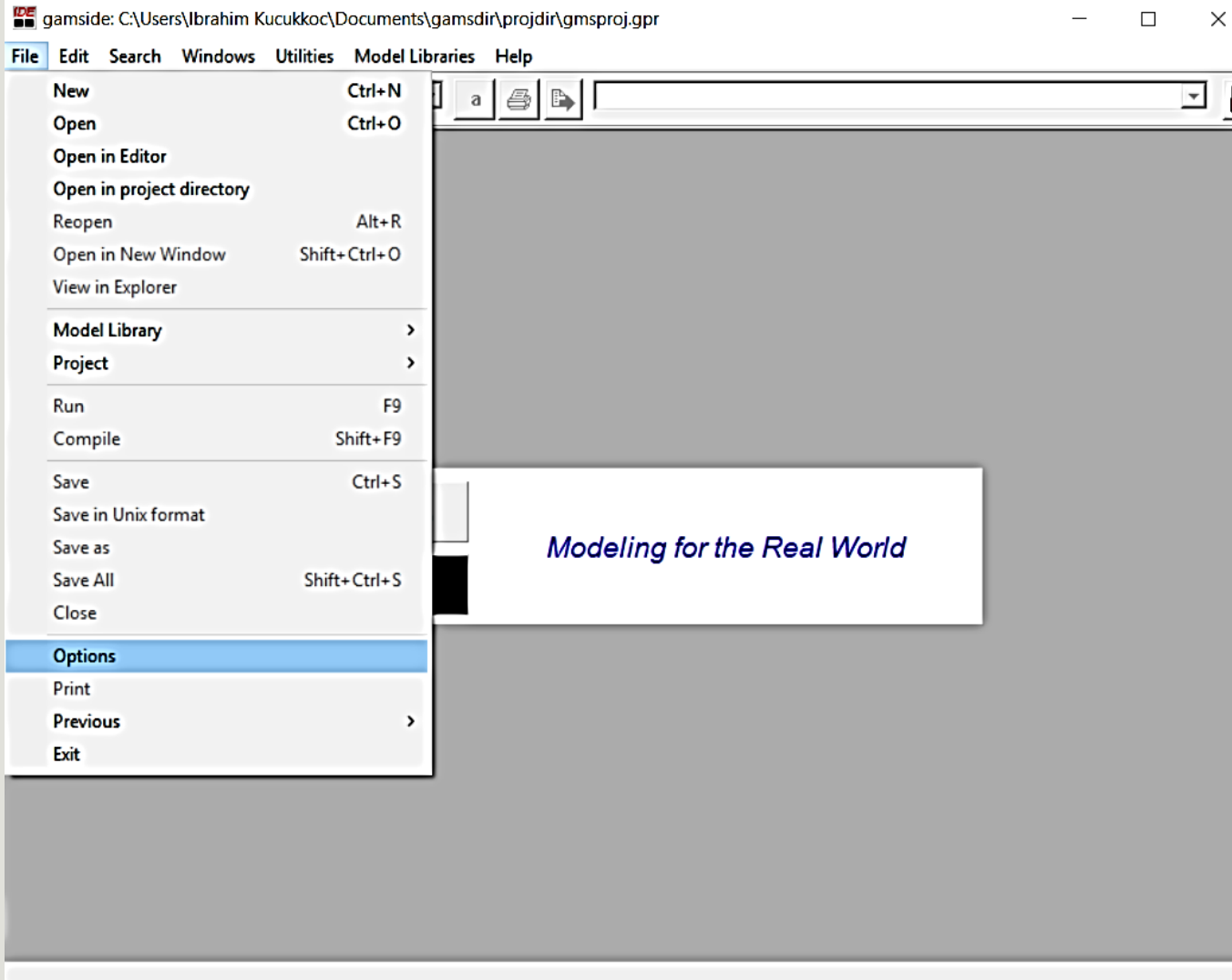


Lisansınızın yapıldığından emin olmak için, “HELP” menüsünden, “About” sekmesine tıklanarak control yapılabilir.

“Current Licence File” alanında gösterilen bilgiler, ürünün lisansının yapıldığını göstermektedir.

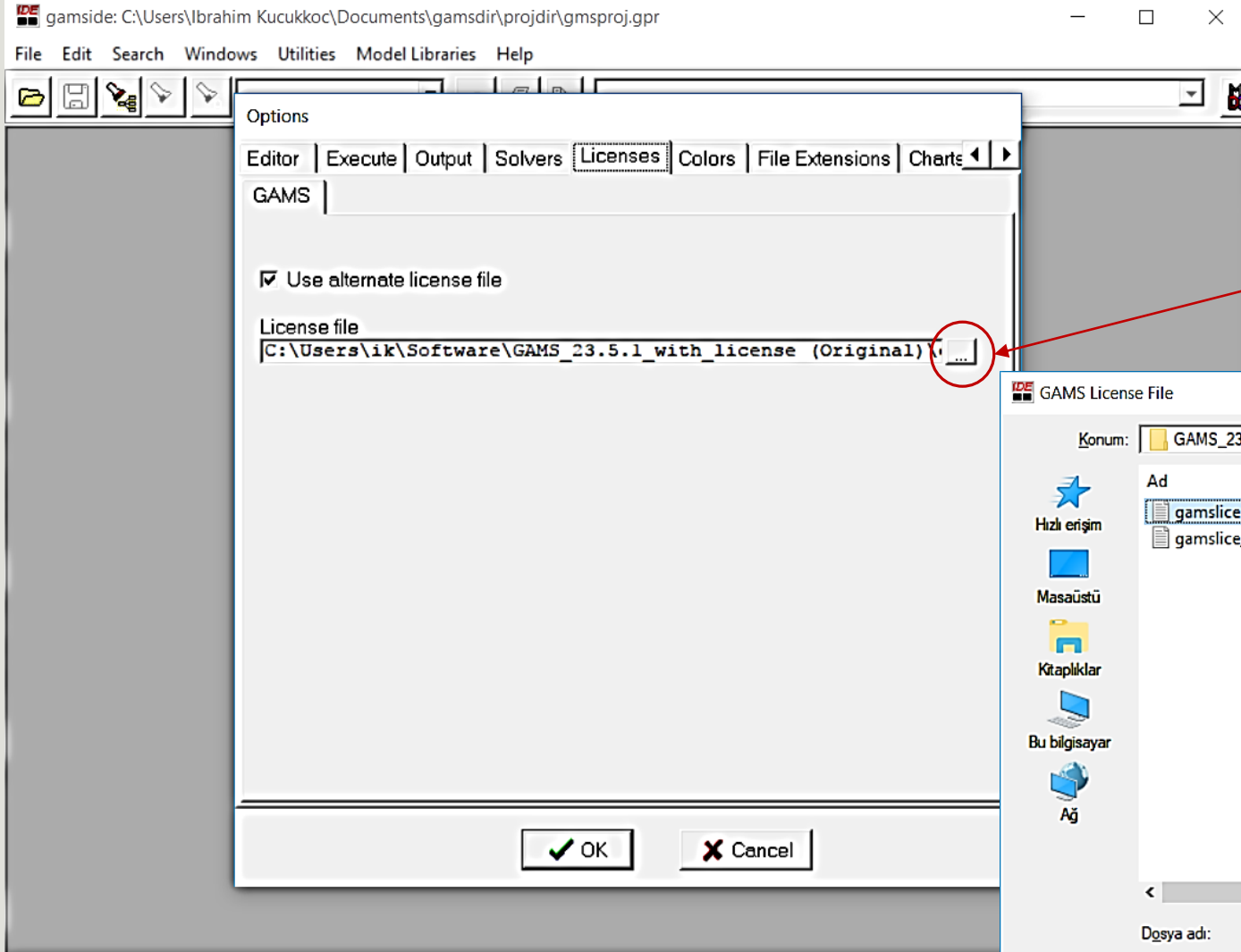
Şayet “Current Licence File” alanı boş ise, ürün lisanslanmamış demektir.

GAMS Kurulumu

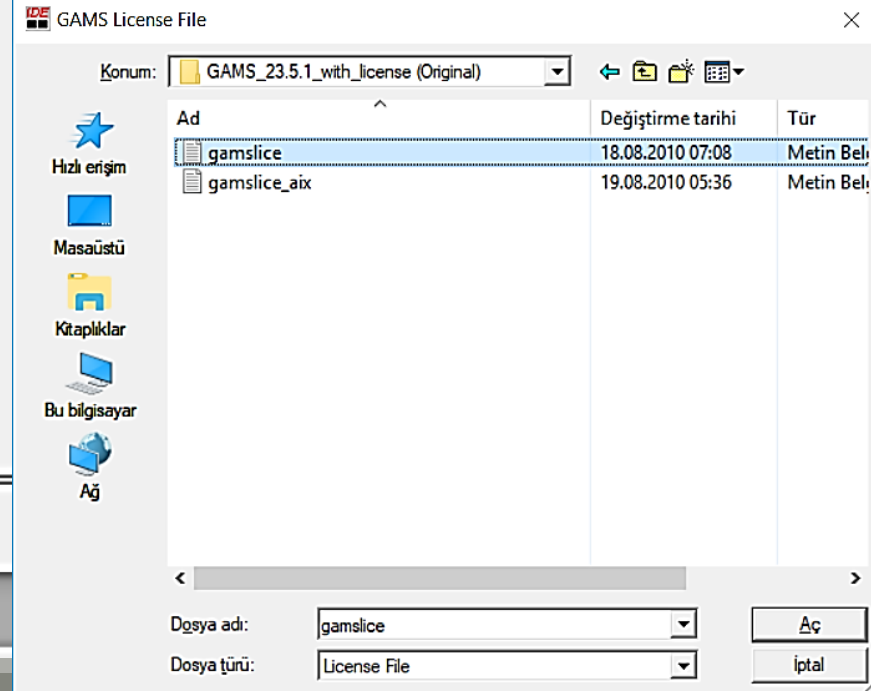


Ürününüzü kurarken lisanslamayı unuttuysanız veya başka bir teknik sebeple lisanslanma başarısız olduysa, “FILE” menüsünden “Options” yolunu izleyerek yeniden lisans dosyanızı tanıtabilirsiniz.

GAMS Kurulumu



Bu alana tıklayarak lisans dosyasının yolunu gösterebilirsiniz.

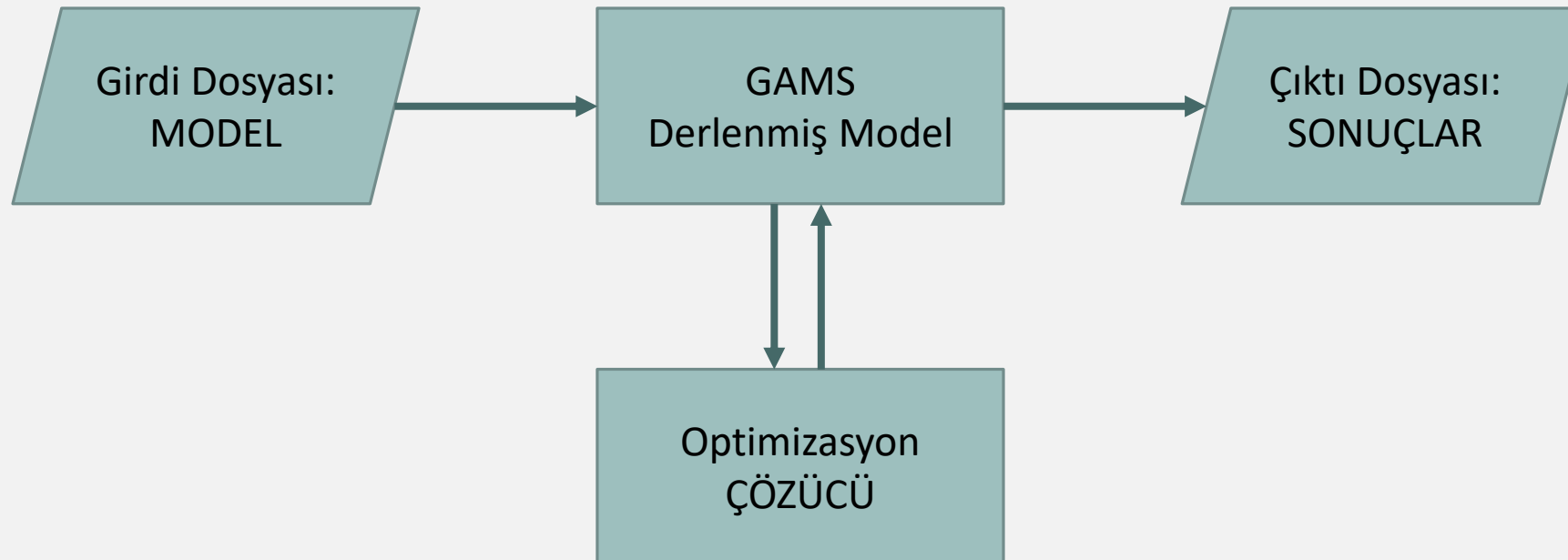


GAMS Nedir?

GAMS (The General Algebraic Modeling System) matematiksel programlama ve optimizasyon için tasarlanan yüksek seviyeli bir dildir.

Doğrusal, Doğrusal Olmayan, Karma Tamsayılı optimizasyon problemleri modellenenebilir.

Büyük, karmaşık problemler modellenenebilir.



GAMS – Örnek 1

Ürün 1 ve ürün 2 olmak üzere 2 farklı ürün imal eden bir firmanın aşağıdaki doğrusal programlama modelini ele aldığımızı varsayalım.

$$\text{Max } Z = 4x_1 + 5x_2$$

$$x_1 + x_2 \leq 60 \quad (\text{I. Numaralı kısıt})$$

$$x_1 - 2x_2 \leq 40 \quad (\text{II. Numaralı kısıt})$$

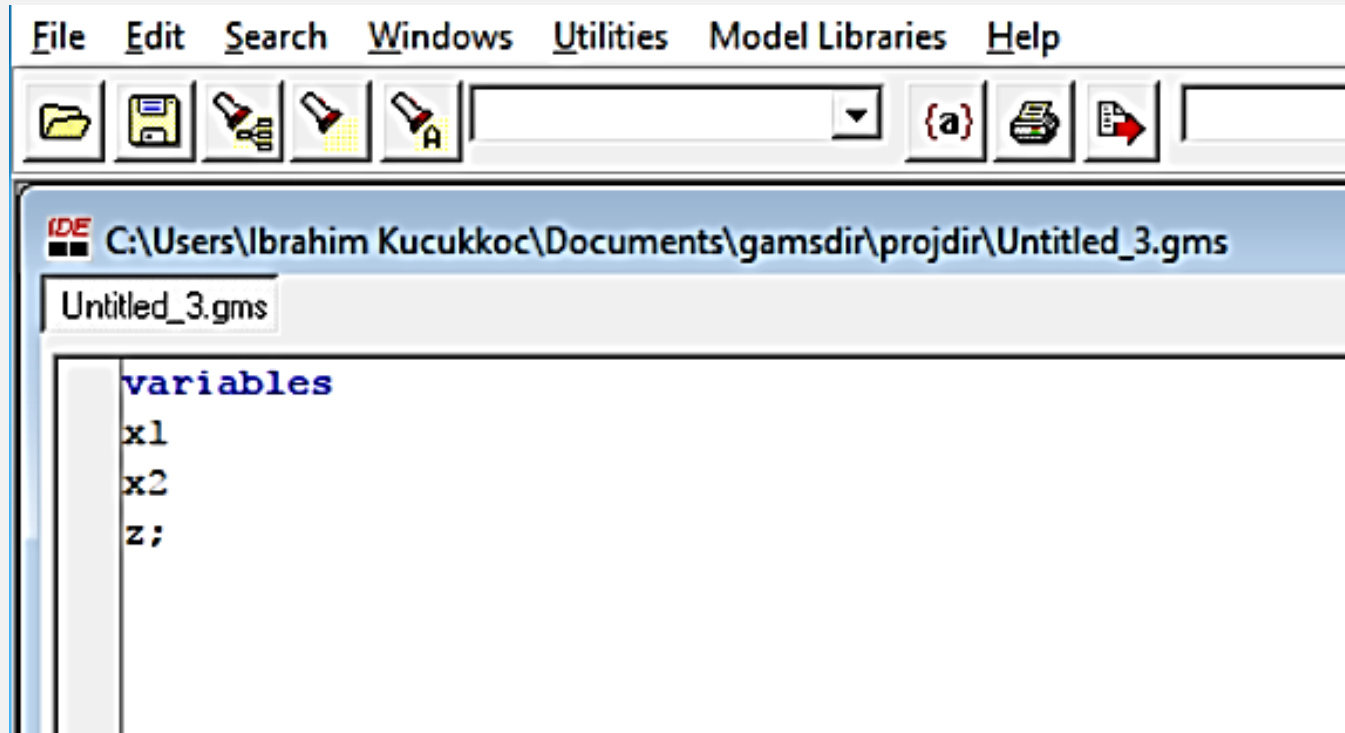
$$x_1 - x_2 \geq 0 \quad (\text{III. Numaralı kısıt})$$

$$x_2 \leq 20 \quad (\text{IV. Numaralı kısıt})$$

$$x_1, x_2 \geq 0$$

GAMS – Örnek 1

Bu modeli GAMS ile modelleyebilmek için önce değişkenlerimizi tanımlamalıyız.

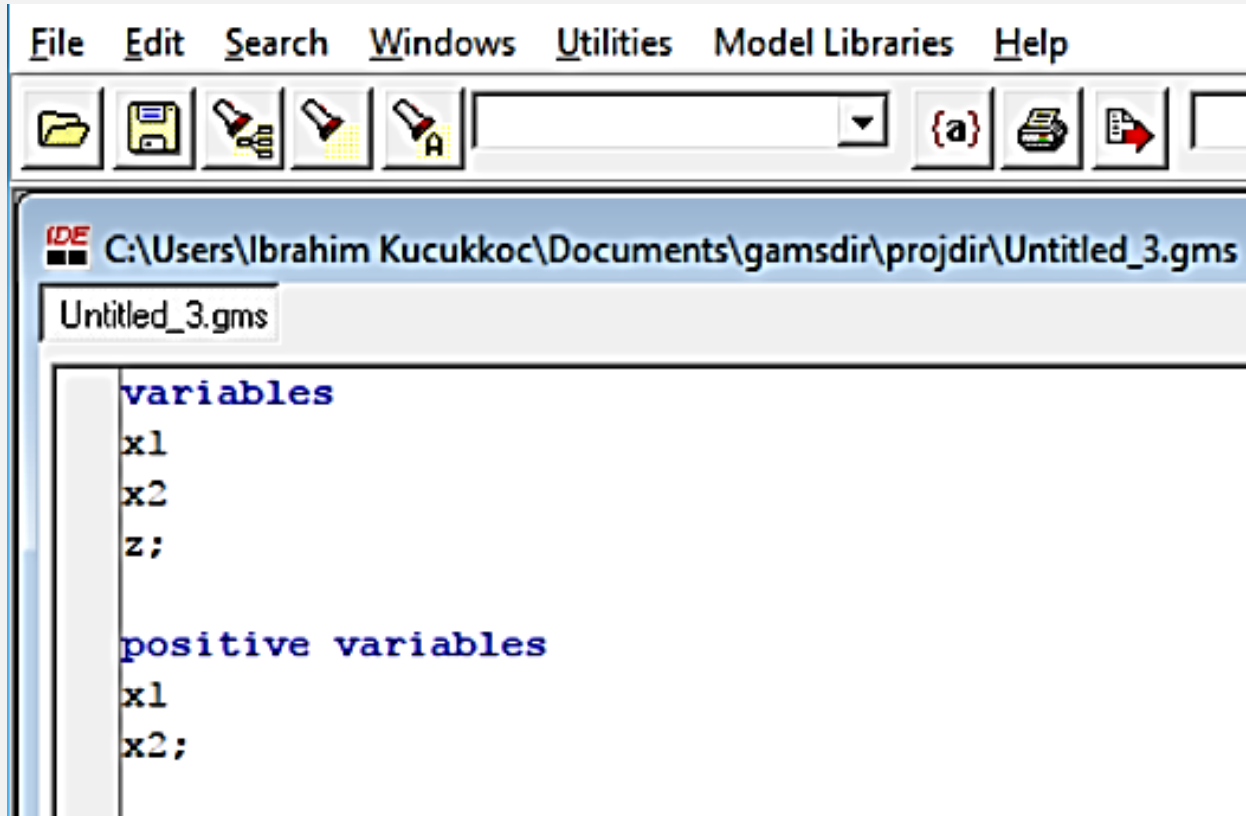


The screenshot shows the GAMS IDE interface. The menu bar includes File, Edit, Search, Windows, Utilities, Model Libraries, and Help. The toolbar contains icons for file operations and model execution. The title bar indicates the file path: C:\Users\Ibrahim Kucukkoc\Documents\gamsdir\projdir\Untitled_3.gms. The editor window shows the following GAMS code:

```
variables  
x1  
x2  
z;
```

GAMS – Örnek 1

Tanımladığımız x_1 ve x_2 değişkenlerinin pozitif olması gereklidir.



The screenshot shows the GAMS IDE interface. The menu bar includes File, Edit, Search, Windows, Utilities, Model Libraries, and Help. The toolbar contains icons for file operations and a command input field. The active window is titled 'Untitled_3.gms' and shows the following code:

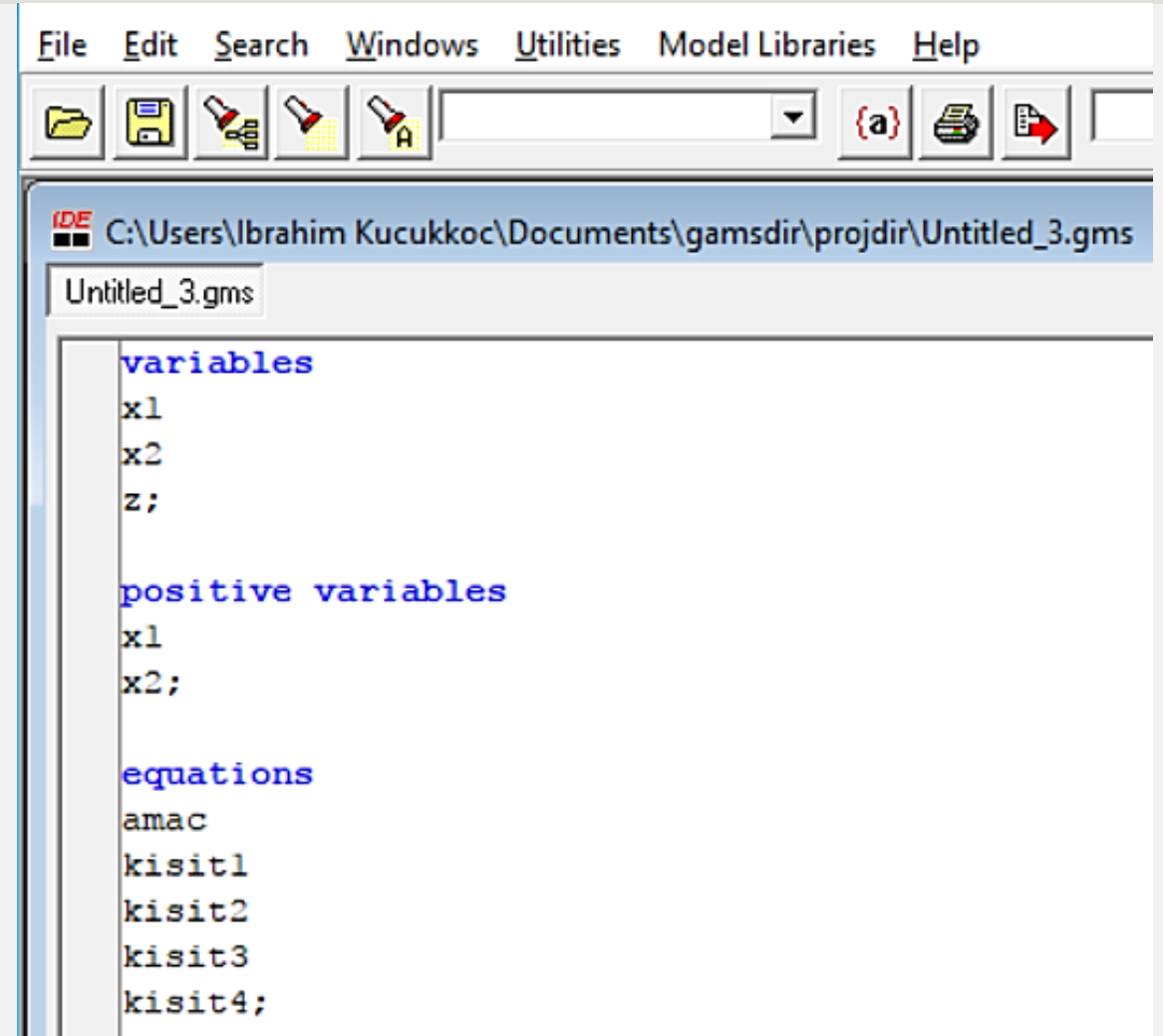
```
variables
x1
x2
z;

positive variables
x1
x2;
```

GAMS – Örnek 1

Daha sonra amaç fonksiyonu ve kısıtları tanımlarız.

Gams kullanırken dikkat etmemiz gereken husus kullanacağımız tüm denklem ve eşitsizlikleri önceden tanımlamamız gerektiğidir.



The screenshot shows the GAMS IDE interface. The menu bar includes File, Edit, Search, Windows, Utilities, Model Libraries, and Help. The toolbar contains icons for file operations and execution. The title bar shows the file path: C:\Users\Ibrahim Kucukkoc\Documents\gamsdir\projdir\Untitled_3.gms. The main window displays the following GAMS code:

```
variables
x1
x2
z;

positive variables
x1
x2;

equations
amac
kisit1
kisit2
kisit3
kisit4;
```

GAMS – Örnek 1

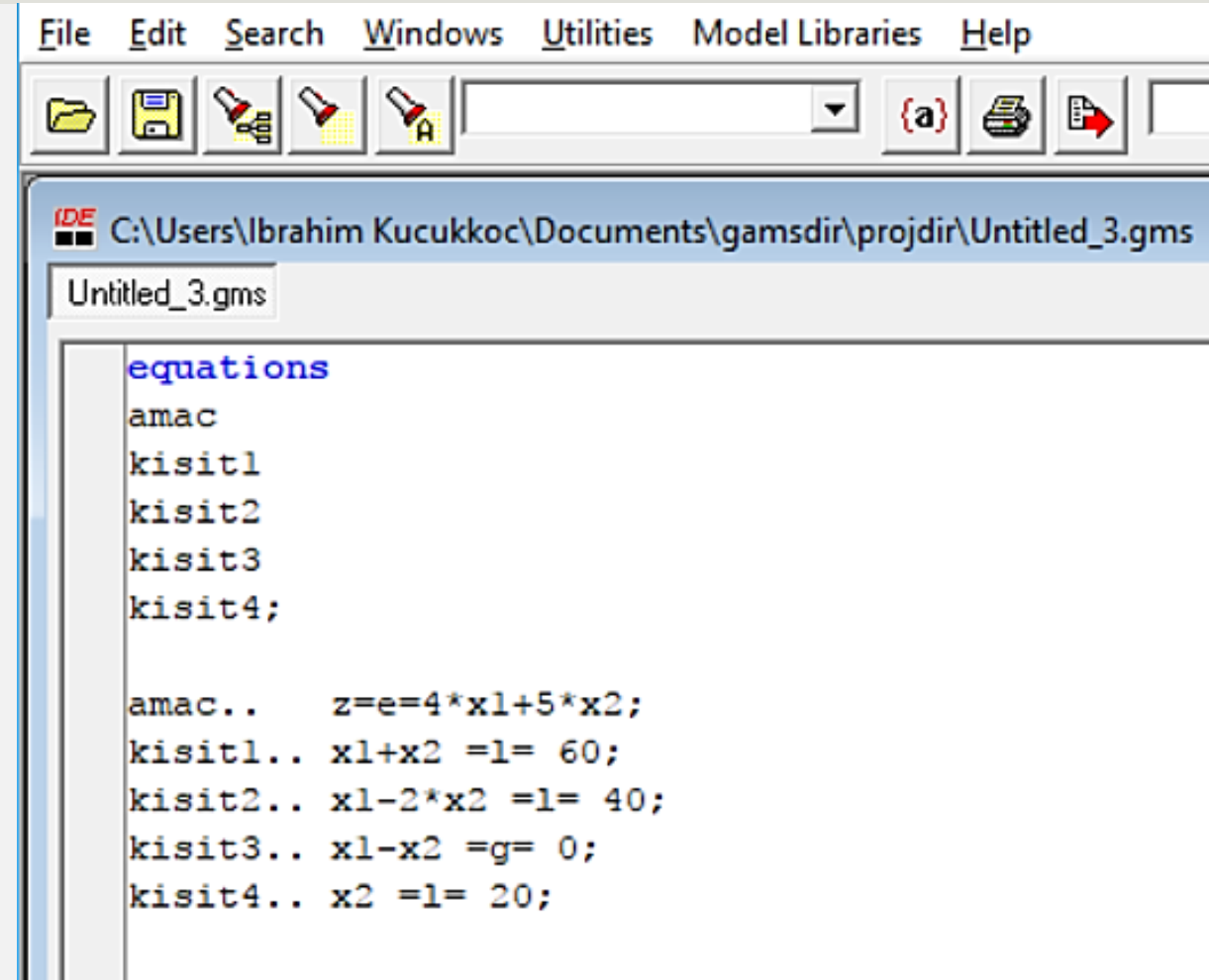
Denklem ve eşitsizlikleri tanımladıktan sonra amaç fonksiyonumuzu ve eşitsizliklerimizi yazmaya başlayabiliriz.

GAMS’de ilişkisel operatörler aşağıdaki şekilde tanımlanır.

$=$ için $= e =$

$\leq$ için $= l =$

$\geq$ için $= g =$

The screenshot shows the GAMS IDE interface. The menu bar includes File, Edit, Search, Windows, Utilities, Model Libraries, and Help. The toolbar contains icons for file operations like opening, saving, and printing. The main window displays the file 'C:\Users\Ibrahim Kucukkoc\Documents\gamsdir\projdir\Untitled_3.gms'. The code in the editor is as follows:

```
equations
amac
kisit1
kisit2
kisit3
kisit4;

amac..    z=e=4*x1+5*x2;
kisit1..  x1+x2 =l= 60;
kisit2..  x1-2*x2 =l= 40;
kisit3..  x1-x2 =g= 0;
kisit4..  x2 =l= 20;
```

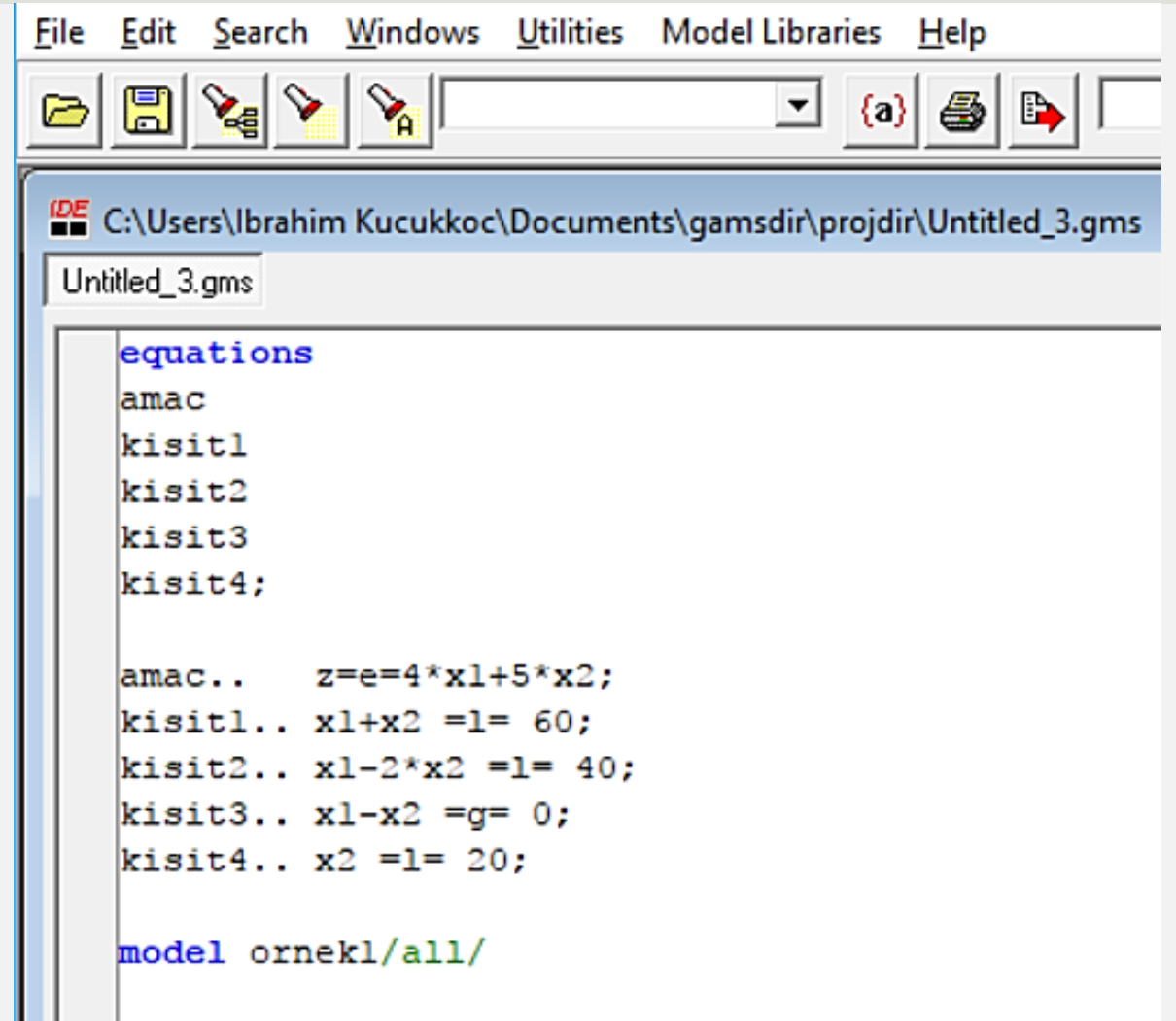
GAMS – Örnek 1

Modelimizi GAMS’de tanımladıktan sonra modelin çözümü için gerekli komutları vermemiz gereklidir.

Bunun için öncelikle modelimize bir isim vermeliyiz.

Örneğimizin adı “ornek1” olursa yandaki gibi modelimizin oluşturulması sağlanabilir.

model ornek1/all/



```
File Edit Search Windows Utilities Model Libraries Help
[Icons: Folder, Save, Print, Find, Run, etc.]
IDE C:\Users\Ibrahim Kucukkoc\Documents\gamsdir\projdir\Untitled_3.gms
Untitled_3.gms

equations
amac
kisit1
kisit2
kisit3
kisit4;

amac..      z=e=4*x1+5*x2;
kisit1..    x1+x2 =l= 60;
kisit2..    x1-2*x2 =l= 40;
kisit3..    x1-x2 =g= 0;
kisit4..    x2 =l= 20;

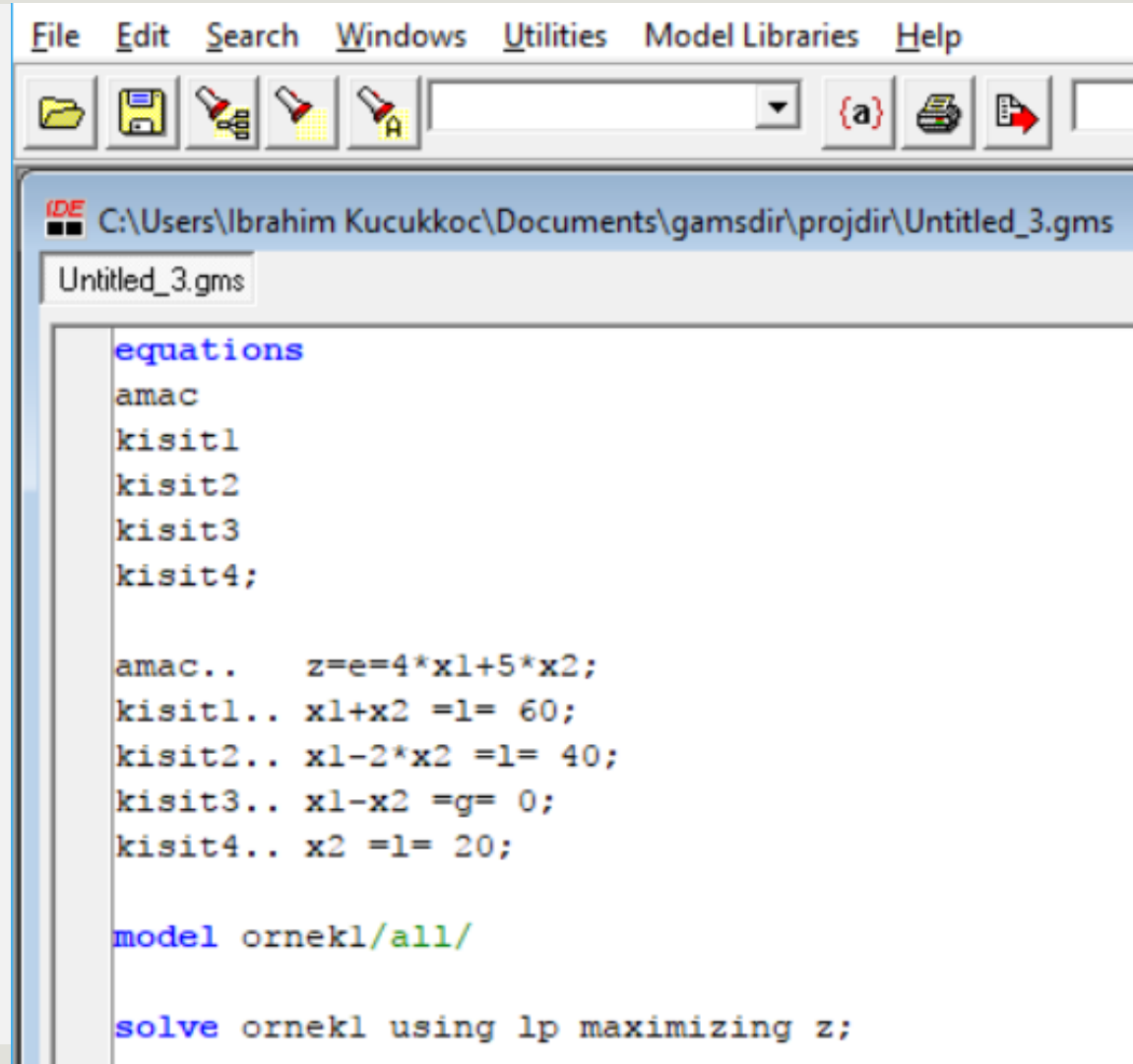
model ornek1/all/
```

GAMS – Örnek 1

- ⦿ Bu komut gereksiz gibi gözükse de aslında ileri seviyede, bir GAMS modeli üstünden bir çok model çalıştırmak isteyen kullanıcılar için çok kullanışlıdır.
- ⦿ “/all/” ifadesindeki “/” işareti ile sınırlandırılan alana, modele dahil edilmesi istenen denklem ve kısıtlar yazılarak model sınırlandırılabilir.
- ⦿ Bu özellik, aynı zamanda her fonksiyonun önce neden tanımlanması gerektiğini açıklamaktadır.
- ⦿ Biz modelimizde tüm kısıtları kullanmak istediğimiz için “all” (tümü) komutunu kullanacağız.

GAMS – Örnek 1

- Son olarak modeli çözmek için “**solve**” komutu çağrılır.
- Bu komutun yapısı aşağıda verilen yapıya uygun olarak oluşturulmalıdır.
 1. “**Solve**” komutu
 2. Çözülmek istenen modelin adı
 3. “**using**” komutu, bu komut kullanmak istediğimiz yöntemi seçim şansı sunar.



The screenshot shows the GAMS IDE interface. The menu bar includes File, Edit, Search, Windows, Utilities, Model Libraries, and Help. The toolbar contains icons for file operations and a search box. The title bar indicates the file path: C:\Users\İbrahim Kucukkoc\Documents\gamsdir\projdir\Untitled_3.gms. The editor window, titled 'Untitled_3.gms', contains the following GAMS code:

```
equations
amac
kisit1
kisit2
kisit3
kisit4;

amac..    z=e=4*x1+5*x2;
kisit1..  x1+x2 =l= 60;
kisit2..  x1-2*x2 =l= 40;
kisit3..  x1-x2 =g= 0;
kisit4..  x2 =l= 20;

model ornek1/all/

solve ornek1 using lp maximizing z;
```


GAMS – Örnek 1

4. Çözüm yöntemi; çözüm yönteminin model yapısına uygun seçilmesi gerekmektedir. GAMS içinde bulunan bazı yöntemler şunlardır.

“lp” doğrusal programlama

“nlp” doğrusal olmayan programlama

“mip” tamsayılı programlama

“rmip” genişletilmiş tamsayılı programlama

“minlp” tamsayılı, doğrusal olmayan programlama

“rminlp” genişletilmiş tamsayılı, doğrusal olmayan programlama

“mpec” denge kısıtlı matematiksel modeller

“cns” kısıtlanmış nonlinear sistemler

GAMS – Örnek 1

5. Amacınıza göre “**minimizing**” veya “**maximizing**” komutu
6. Optimize edilmek istenen değişkenin adı (bu örnekte “**z**”)

W2Ornek1.gms

Kodun Tamamı:

variables

x1
x2
z;

positive variables

x1
x2;

equations

amac
kisit1
kisit2
kisit3
kisit4;

amac.. z=e=4*x1+5*x2;
kisit1.. x1+x2 =l= 60;
kisit2.. x1-2*x2 =l= 40;
kisit3.. x1-x2 =g= 0;
kisit4.. x2 =l= 20;

model ornek1/all/

solve ornek1 using lp maximizing z;



C:\Users\İbrahim Kucukkoc\Documents\gamsdir\projdir\W1Ornek1.lst

W1Ornek1.gms Untitled_3.gms Untitled_4.gms Untitled_4.lst Untitled_5.lst W1Ornek1.lst

- Compilation
 - Equation Listing SOLVE ornek1 Using L
 - Equation
 - Column Listing SOLVE ornek1 Using L
 - Column
 - Model Statistics SOLVE ornek1 Using Lf
 - Solution Report SOLVE ornek1 Using LI
 - SoIEQU
 - SoIVAR

S O L V E S U M M A R Y

MODEL ornek1 OBJECTIVE z
TYPE LP DIRECTION MAXIMIZE
SOLVER CPLEX FROM LINE 25

**** SOLVER STATUS 1 Normal Completion
**** MODEL STATUS 1 Optimal
**** OBJECTIVE VALUE 260.0000

RESOURCE USAGE, LIMIT 0.003 1000.000
ITERATION COUNT, LIMIT 1 2000000000

IBM ILOG CPLEX Jul 4, 2010 23.5.1 WIN 18414.18495 VS8 x86/MS Windows
Cplex 12.2.0.0, GAMS Link 34

LP status(1): optimal
Optimal solution found.
Objective : 260.000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	60.000	60.000	4.000
---- EQU kisit2	-INF	.	40.000	.
---- EQU kisit3	.	20.000	+INF	.
---- EQU kisit4	-INF	20.000	20.000	1.000

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR x1	.	40.000	+INF	.
---- VAR x2	.	20.000	+INF	.
---- VAR z	-INF	260.000	+INF	.

No active process

untitled_4 untitled_5 w1ornek1

--- W1Ornek1.gms(25) 3 Mb
--- 5 rows 3 columns 10 non-zeroes
--- Executing CPLEX: elapsed 0:00:00.077

IBM ILOG CPLEX Jul 4, 2010 23.5.1 WIN 18414.18495 VS8
Cplex 12.2.0.0, GAMS Link 34

Reading data...
Starting Cplex...
Tried aggregator 1 time.
LP Presolve eliminated 2 rows and 1 columns.
Reduced LP has 3 rows, 2 columns, and 6 nonzeros.
Presolve time = 0.00 sec.

Iteration	Dual Objective	In Variable
1 I	0.000000	x1

LP status(1): optimal

Optimal solution found.
Objective : 260.000000

--- Restarting execution
--- W1Ornek1.gms(25) 0 Mb
--- Reading solution for model ornek1
*** Status: Normal completion
--- Job W1Ornek1.gms Stop 02/07/18 19:07:42 elapsed 0:00:0

Close Open Log ☐ Summary only ☒ Update

Örnek 2

ABC firması, standart ve lüks bilgisayar olarak adlandırdığı iki tip bilgisayar üretmektedir.

Standart bilgisayarlar için normal kapasiteli, lüks bilgisayarlar içinse yüksek kapasiteli sabit disk (SD) kullanılmaktadır. Ayrıca standart bilgisayarlarda bir adet, lüks bilgisayarlarda iki adet RAM vardır.

Firma, standart bilgisayar başına \$30, lüks bilgisayar başına \$50 kar etmektedir.

Bir ayda en çok 60 adet normal kapasiteli SD, 50 adet yüksek kapasiteli SD ve 120 adet RAM kullanılabilir.

Firma, stok değerlerini aşmayacak şekilde iki tip bilgisayardan ayda kaç adet üretmesi durumunda kar maksimize edilecektir?

Firmanın aylık standart bilgisayar üretim miktarı X_s , aylık lüks bilgisayar üretimi X_l ile gösterilirse; amaç karı maksimize etmek olduğundan amaç fonksiyonu:

$$\text{Max: } 30X_s + 50X_l$$

Kısıtlar:

$$X_s \leq 60$$

$$X_l \leq 50$$

$$X_s + 2X_l \leq 120$$

$$X_s, X_l \geq 0$$

W3Ornek2.gms

```
variables
xs, xl
z;

positive variables
xs, xl;

equations
amac
kisit1
kisit2
kisit3
;

amac..    z =e= 30*xs+50*xl;
kisit1..  xs =l= 60;
kisit2..  xl =l= 50;
kisit3..  xs + 2*xl =l= 120;

model ornek3 /all/;
solve ornek3 using lp maximizing z;
```

Reading data...
 Starting Cplex...
 Tried aggregator 1 time.
 LP Presolve eliminated 3 rows and 1 columns.
 Reduced LP has 1 rows, 2 columns, and 2 nonzeros.
 Presolve time = 0.00 sec.

Iteration	Dual Objective	In Variable
1	3300.000000	x1

LP status(1): optimal

Optimal solution found.
 Objective : 3300.000000

S O L V E S U M M A R Y

MODEL	ornek3	OBJECTIVE	z
TYPE	LP	DIRECTION	MAXIMIZE
SOLVER	CPLEX	FROM LINE	21

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 3300.0000

RESOURCE USAGE, LIMIT	0.035	1000.000
ITERATION COUNT, LIMIT	1	2000000000

LP status(1): optimal
 Optimal solution found.
 Objective : 3300.000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	60.000	60.000	5.000
---- EQU kisit2	-INF	30.000	50.000	.
---- EQU kisit3	-INF	120.000	120.000	25.000

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR xs	.	60.000	+INF	.
---- VAR x1	.	30.000	+INF	.
---- VAR z	-INF	3300.000	+INF	.

Ödev 1

Aşağıdaki problemi GAMS ile çözünüz.

$$\max P = 20x_1 + 10x_2 + 15x_3$$

Subject to

$$3x_1 + 2x_2 + 5x_3 \leq 55$$

$$2x_1 + x_2 + x_3 \leq 26$$

$$x_1 + x_2 + 3x_3 \leq 30$$

$$5x_1 + 2x_2 + 4x_3 \leq 57$$

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

GAMS - Sets, variables, parameters, tables, equations, model, solve

4

Bir GAMS Modelinin Temel Bileşenleri

Inputs:

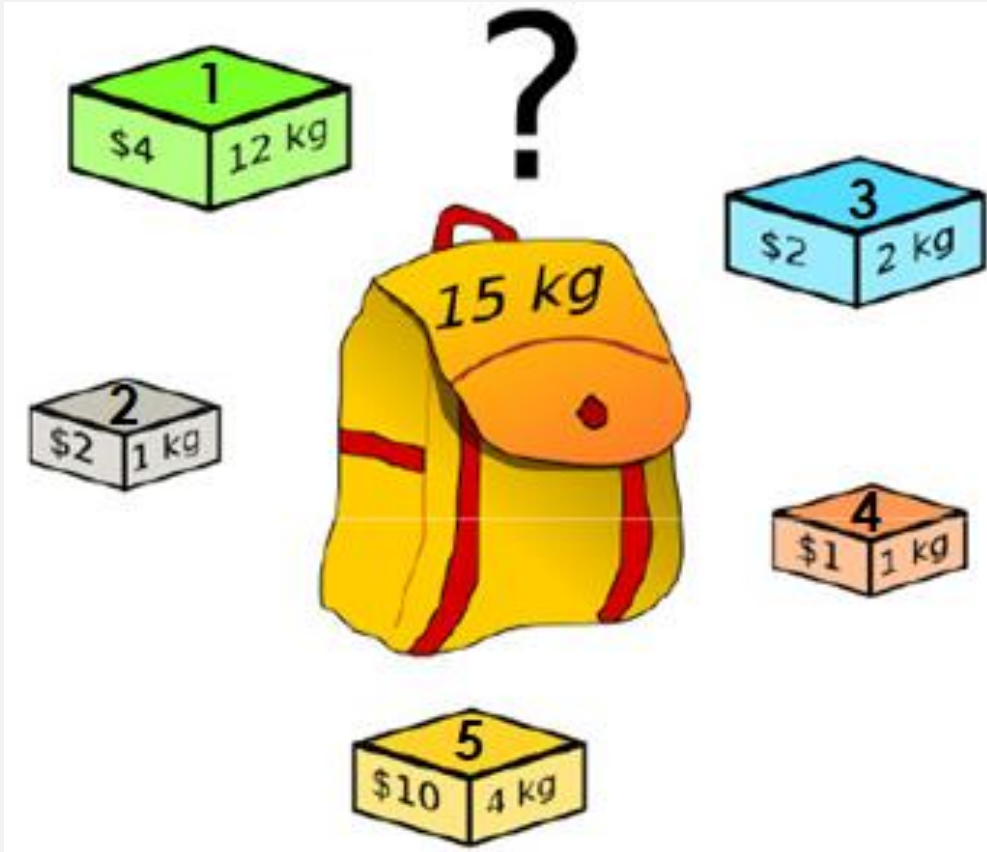
- Sets
 - Declaration
 - Assignment of members
- Data (Parameters, Tables, Scalars)
 - Declaration
 - Assignment of values
- Variables
 - Declaration
 - Assignment of type
- Assignment of bounds and/or initial values (optional)
- Equations
 - Declaration
 - Definition
- Model and Solve statements
- Display statement (optional)

Outputs:

- Echo Print
- Reference Maps
- Equation Listings
- Status Reports
- Results

Sırt Çantası (Knapsack) Problemi Örneği

Sırt çantası (knapsack) problemi, sınırlı olarak verilen kapasite kısıtını sağlarken faydayı maksimize etmeye çalışma problemidir.



n : eşya sayısı ($i = 1, 2, \dots, n$)

w_i : i eşyasının ağırlığı (kg)

v_i : i eşyasının faydası (\$)

W : Çantanın kapasitesi (kg)

X_i : 1, eğer i eşyası alınmışsa; 0, alınmamışsa

$$\max \sum_i v_i * X_i, \quad i = 1, 2, \dots, n$$

Kapasite Kısıtı:
$$\sum_i w_i * X_i \leq W$$

İşaret Kısıtı:
$$X_i \in \{0, 1\}$$

```
set
  i eşya numarası /1*5/
;
* i eşya numarası /1,2,3,4,5/

parameters
  we(i) eşya ağırlığı
  /1 12
   2 1
   3 2
   4 1
   5 4/

  v(i) eşyanın değeri
  /1 4
   2 2
   3 2
   4 1
   5 10/
;

scalar W çanta kapasitesi /15/;
```

```
variables
  X(i) karar değişkeni
  Z amaç fonk
;

binary variables
X(i)
;

equations
  amac
  kisitl
;

amac.. Z =e= sum(i, v(i)*X(i));
kisitl.. sum(i, we(i)*X(i)) =l= W;

model knapsack /all/;
solve knapsack using mip maximizing Z;
```

Sum kullanmadan yapılmak istenirse, amaç fonksiyonu ve kısıt aşağıdaki şekilde de tanımlanabilir:

```
equations
amac
kisit1
;

amac.. Z =e= v("1")*X('1')+v("2")*X('2')+v("3")*X('3')+v("4")*X('4')+v("5")*X('5');
kisit1.. we("1")*X('1')+we("2")*X('2')+we("3")*X('3')+we("4")*X('4')+we("5")*X('5') =1= W;

model knapsack /all/;
solve knapsack using mip maximizing Z;
```

S O L V E S U M M A R Y

MODEL knapsack OBJECTIVE Z
TYPE MIP DIRECTION MAXIMIZE
SOLVER CPLEX FROM LINE 42

**** SOLVER STATUS 1 Normal Completion
**** MODEL STATUS 1 Optimal
**** OBJECTIVE VALUE 15.0000

RESOURCE USAGE, LIMIT 0.056 1000.000
ITERATION COUNT, LIMIT 1 2000000000

Görüleceği üzere X2, X3, X4 ve X5 karar değişkenleri 1 değerini almışlardır.

Bu durumda amaç fonksiyonu değeri \$15 olmuştur.

Çanta ağırlığı da 8 kg'dır.

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	8.000	15.000	.

---- VAR X karar değişkeni

	LOWER	LEVEL	UPPER	MARGINAL
1	.	.	1.000	4.000
2	.	1.000	1.000	2.000
3	.	1.000	1.000	2.000
4	.	1.000	1.000	1.000
5	.	1.000	1.000	10.000

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR Z	-INF	15.000	+INF	.

Z amaç fonk

Bütçe Kısıtlı Proje Seçimi

Bir şirket gelecek yıl için sekiz olası proje yatırımını değerlendirmektedir. Her bir projenin beklenen getirisi ve her bir proje için yapılması gereken başlangıç yatırımı tabloda verilmiştir. Bütçeyi (\$1.5 milyon) aşmayarak karı maksimize edecek şekilde hangi projelerin yatırıma dahil edilmesi gerektiğine karar veriniz.

Proje No	Beklenen Getiri	Baslangic Yatirimi
1	\$500.000	\$425.000
2	\$750.000	\$450.000
3	\$1.000.000	\$550.000
4	\$600.000	\$300.000
5	\$500.000	\$150.000
6	\$500.000	\$250.000
7	\$450.000	\$350.000
8	\$500.000	\$325.000

*Bütçe Kısıtlı Proje Seçimi

sets

```
i yatırım numarası /1*8/
;
```

parameters

```
g(i) beklenen getiri
/1 500
 2 750
 3 1000
 4 600
 5 500
 6 500
 7 450
 8 500/
```

y(i) başlangıç yatırımı

```
/1 425
 2 450
 3 550
 4 300
 5 150
 6 250
 7 350
 8 325/
```

```
;
```

```
scalar B bütçe /1500/;
```

variable

```
x(i) eğer i projesine yatırım yapılmışsa 1 değilse 0
z amaç fonk değeri
;
```

binary variable

```
x(i)
;
```

equations

```
amac
kisit bütçe yatırımı
;
```

```
amac.. z=e=sum(i, x(i)*(g(i)-y(i)));
```

```
kisit.. sum(i, x(i)*y(i)) =l= B;
```

```
model butce /all/;
```

```
solve butce using mip maximizing z;
```


S O L V E S U M M A R Y

MODEL butce OBJECTIVE z
 TYPE MIP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 49

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 8 Integer Solution
 **** OBJECTIVE VALUE 1400.0000

MIP Solution: 1400.000000 (1 iterations, 0 nodes)
 Final Solve: 1400.000000 (0 iterations)
 Best possible: 1516.666667
 Absolute gap: 116.666667
 Relative gap: 0.083333

LOWER LEVEL UPPER MARGINAL

---- EQU amac . . . 1.000
 ---- EQU kisit -INF 1450.000 1500.000 .

kisit bütçe yatırımı

---- VAR x eğer i projesine yatırım yapılmışsa 1 değilse 0

LOWER LEVEL UPPER MARGINAL

1	.	.	1.000	75.000
2	.	1.000	1.000	300.000
3	.	1.000	1.000	450.000
4	.	1.000	1.000	300.000
5	.	1.000	1.000	350.000
6	.	.	1.000	250.000
7	.	.	1.000	100.000
8	.	.	1.000	175.000

Probleme aşağıdaki gibi yeni bir kısıt daha ekleyelim:
 - İlk dört projeye yapılabilecek yatırım tutarı \$1milyon'u aşamaz.

*Bütçe Kısıtlı Proje Seçimi

*Yeni Kısıt: İlk dört projeye yapılacak yatırım tutarı 1000 (\$K) 'ı geçemez!

sets

i yatırım numarası /1*8/;

parameters

g(i) beklenen getiri (\$K)

```
/1 500
2 750
3 1000
4 600
5 500
6 500
7 450
8 500/
```

y(i) başlangıç yatırımı (\$K)

```
/1 425
2 450
3 550
4 300
5 150
6 250
7 350
8 225/
```

scalar B bütçe (\$K) /1500/;

scalar V ilk dört proje için bütçe /1000/;

variable

x(i) eğer i projesine yatırım yapılmışsa 1 değilse 0

z amaç fonk değeri

;

binary variable

x(i)

;

equations

amac

kisit toplam bütçe kısıtı

kisit2 ilk dört proje için bütçe kısıtı

;

amac.. z=e=sum(i, x(i)*(g(i)-y(i)));

kisit.. sum(i, x(i)*y(i)) =l= B;

*kisit2.. x('1')*y('1')+ x('2')*y('2')+x('3')*y('3')+x('4')*y('4') =l= V;

kisit2.. sum(i\$(ord(i)<=4), x(i)*y(i)) =l= V;

model butce /all/;

solve butce using mip maximizing z;

S O L V E S U M M A R Y

MODEL butce OBJECTIVE z
TYPE MIP DIRECTION MAXIMIZE
SOLVER CPLEX FROM LINE 52

**** SOLVER STATUS 1 Normal Completion
**** MODEL STATUS 8 Integer Solution
**** OBJECTIVE VALUE 1350.0000

MIP Solution: 1350.000000 (2 iterations, 0 nodes)
Final Solve: 1350.000000 (0 iterations)

Best possible: 1484.615385
Absolute gap: 134.615385
Relative gap: 0.099715

LOWER LEVEL UPPER MARGINAL

---- EQU amac . . . 1.000
---- EQU kisit -INF 1250.000 1500.000 .
---- EQU kisit2 -INF 850.000 1000.000 .

kisit toplam bütçe kısıtı
kisit2 ilk dört proje için bütçe kısıtı

---- VAR x eğer i projesine yatırım yapılmışsa 1 değilse 0

LOWER LEVEL UPPER MARGINAL

1	.	.	1.000	75.000
2	.	.	1.000	300.000
3	.	1.000	1.000	450.000
4	.	1.000	1.000	300.000
5	.	1.000	1.000	350.000
6	.	1.000	1.000	250.000
7	.	.	1.000	100.000
8	.	.	1.000	175.000

Ahşap Üretim Problemi

Dört tip ahşap panelin (A, B, C, D) üretildiği bir sistemi ele alalım. Her ahşap panel, çam ve meşe talaşlarının farklı miktarlarda karıştırılıp yapıştırılması ve preslenmesi sonucu oluşmaktadır.

Aşağıdaki tablo, değişik panel tipleri için bir palet panel üretiminde kullanılması gereken kaynak miktarlarını göstermektedir.

Her Bir Palet İçin Gerekli Kaynak Miktarı				
	A	B	C	D
Yapıştırıcı (litre) – $y(i)$	50	50	100	50
Sıkıştırma (saat) – $s(i)$	5	15	10	5
Çam Talaşı (kg) – $c(i)$	500	400	300	200
Meşe Talaşı (kg) – $m(i)$	500	750	250	500

Eldeki (mevcut) kaynak miktarları şu şekildedir:

5800 litre yapıştırıcı (Y)

730 saat sıkıştırma kapasitesi (S)

29200 kg çam talaşı (C)

60500 kg meşe talaşı (M)

Her bir palet A, B, C ve D paneli sırasıyla \$450, \$1150, \$800 ve \$400 dolara satılmaktadır.

Firma karını maksimize etmek için her bir panel tipinden kaç palet üretmelidir?

Öncelikle problem matematiksel model olarak ifade edilmelidir.

Karar değişkenleri A, B, C ve D modellerinin üretim miktarlarıdır. Bunlar sırasıyla X_1 , X_2 , X_3 ve X_4 olarak düşünülürse amaç fonksiyonu aşağıdaki şekilde yazılabilir:

$$\text{Max } Z = 450X_1 + 1150X_2 + 800X_3 + 400X_4$$

Kaynak kullanımlarına ilişkin kısıtlar aşağıdaki şekilde ifade edilebilir:

$$50 X_1 + 50 X_2 + 100 X_3 + 50 X_4 \leq 5800 \text{ (yapıştırıcı)}$$

$$5 X_1 + 15 X_2 + 10 X_3 + 5 X_4 \leq 730 \text{ (sıkıştırma)}$$

$$500 X_1 + 400 X_2 + 300 X_3 + 200 X_4 \leq 29200 \text{ (çam)}$$

$$500 X_1 + 750 X_2 + 250 X_3 + 500 X_4 \leq 60500 \text{ (meşe)}$$

Herhangi bir panelin üretim miktarı 0'dan küçük olamayacağı için:

$$X_1, X_2, X_3, X_4 \geq 0$$

```
variables
x1, x2, x3, x4
z;

positive variables
x1, x2, x3, x4;

equations
amac
kisit1
kisit2
kisit3
kisit4
;

amac..    z =e= 450*x1+1150*x2+800*x3+400*x4;
kisit1..  50*x1 + 50*x2 + 100*x3 + 50*x4 =l= 5800;
kisit2..  5*x1 + 15*x2 + 10*x3 + 5*x4 =l= 730;
kisit3..  500*x1 + 400*x2 + 300*x3 + 200*x4 =l= 29200;
kisit4..  500*x1 + 750*x2 + 250*x3 + 500*x4 =l= 60500;

model ornek3 /all/;
solve ornek3 using lp maximizing z;
```

S O L V E S U M M A R Y

MODEL ornek3 OBJECTIVE z
TYPE LP DIRECTION MAXIMIZE
SOLVER CPLEX FROM LINE 23

**** SOLVER STATUS 1 Normal Completion
**** MODEL STATUS 1 Optimal
**** OBJECTIVE VALUE 58800.0000

RESOURCE USAGE, LIMIT 0.005 1000.000
ITERATION COUNT, LIMIT 4 2000000000

LP status(1): optimal
Optimal solution found.
Objective : 58800.000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	5800.000	5800.000	0.429
---- EQU kisit2	-INF	730.000	730.000	71.429
---- EQU kisit3	-INF	29200.000	29200.000	0.143
---- EQU kisit4	-INF	32500.000	60500.000	.

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR x1	.	23.000	+INF	.
---- VAR x2	.	15.000	+INF	.
---- VAR x3	.	39.000	+INF	.
---- VAR x4	.	.	+INF	-7.143
---- VAR z	-INF	58800.000	+INF	.

Modeli bu şekilde açık olarak yazmak doğru olmasına rağmen, bu yaklaşım kullanıldığında büyük kapsamlı modellerin yazılmasında problemler çıkabilir. Bu sebeple yazım üstünde bazı değişiklikler yapılabilir.

Modelde verilen dört karar değişkeninin ayrı ayrı tanımlanmasına gerek yoktur. Bunun yerine bir x değişkeni tanımlanıp, dört ayrı **indis** atanabilir.

GAMS modellerinde indislerin “**set**” komutu ile tanımlanabileceği gösterilmişti.

set i /1, 2, 3, 4/

Bu komut aynı sonucu verecek şekilde aşağıdaki gibi de yazılabilir:

set i /1*4/

Ayrıca parametreler ve scalar değerler de kullanılabilir.

Bu model kapalı formda aşağıdaki şekilde yazılabilir:

Amaç Fonksiyonu:

$$\max Z = \sum_{i \in I} p_i X_i$$

Kısıtlar:

$$\sum_{i \in I} y_i X_i \leq Y$$

$$\sum_{i \in I} s_i X_i \leq S$$

$$\sum_{i \in I} c_i X_i \leq C$$

$$\sum_{i \in I} m_i X_i \leq M$$

$$X_i \geq 0, \quad \forall i \in I$$

Notasyon:

I : Ürün kümesi (A, B, C, D)

p_i : Bir palet i ürünü satışından elde edilecek gelir

y_i : Bir palet i ürünü için gerekli yapıştırıcı miktarı

s_i : Bir palet i ürünü için gerekli sıkıştırma kapasitesi kullanım miktarı

c_i : Bir palet i ürünü için gerekli çam talaşı miktarı

m_i : Bir palet i ürünü için gerekli meşe talaşı miktarı

X_i : i ürününden üretilecek miktar (palet)

Y : Mevcut yapıştırıcı kapasitesi

S : Mevcut sıkıştırma kapasitesi

C : Mevcut çam talaşı kapasitesi

M : Mevcut meşe talaşı kapasitesi

```

sets
i /1*4/

variables
x(i)
z;

positive variables
x(i);

parameters
y(i) yapıştırıcı
/ 1 50
  2 50
  3 100
  4 50/

s(i) yapıştırıcı
/ 1 5
  2 15
  3 10
  4 5/

c(i) yapıştırıcı
/ 1 500
  2 400
  3 300
  4 200/

m(i) yapıştırıcı
/ 1 500
  2 750
  3 250
  4 500/

```

```

p(i) yapıştırıcı
/ 1 450
  2 1150
  3 800
  4 400/
;

scalars
YK /5800/
SK /730/
CK /29200/
MK /60500/
;

equations
amac
kisit1
kisit2
kisit3
kisit4
;

amac.. z =e= sum(i, p(i)*x(i));
kisit1.. sum(i, y(i)*x(i)) =l= YK;
kisit2.. sum(i, s(i)*x(i)) =l= SK;
kisit3.. sum(i, c(i)*x(i)) =l= CK;
kisit4.. sum(i, m(i)*x(i)) =l= MK;

model ornek3 /all/;
solve ornek3 using lp maximizing z;

```

S O L V E S U M M A R Y

MODEL ornek3 OBJECTIVE z
 TYPE LP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 65

LP status(1): optimal
 Optimal solution found.
 Objective : 58800.000000

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 58800.0000

RESOURCE USAGE, LIMIT 0.003 1000.000
 ITERATION COUNT, LIMIT 4 20000000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	5800.000	5800.000	0.429
---- EQU kisit2	-INF	730.000	730.000	71.429
---- EQU kisit3	-INF	29200.000	29200.000	0.143
---- EQU kisit4	-INF	32500.000	60500.000	.

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1	.	23.000	+INF	.
2	.	15.000	+INF	.
3	.	39.000	+INF	.
4	.	.	+INF	-7.143

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR z	-INF	58800.000	+INF	.

Ulaştırma Problemi

İyi bilinen bir taşıma (ulaştırma) probleminde, belirli arz noktalarından, belirli talep noktalarına taşınacak ürünlerin minimum maliyetle taşınması istenmektedir. Şehirler arası uzaklıklar bilinmektedir. Problem, taşıma maliyetini minimize etmek amacıyla, herhangi bir arz noktası-talep noktası arasında ne kadar malzeme taşınacağına belirlenmesi sürecidir.

Indices:

i = plants

j = markets

Given Data:

a_i = supply of commodity of plant i (in cases)

b_j = demand for commodity at market j

c_{ij} = cost per unit shipment between plant i and market j (\$/case)

Decision Variables:

x_{ij} = amount of commodity to ship from plant i to market j (cases),

where $x_{ij} \geq 0$, for all i, j

Constraints:

Observe supply limit at plant i : $\sum_j x_{ij} \leq a_i$ for all i (cases)

Satisfy demand at market j : $\sum_i x_{ij} \geq b_j$ for all j (cases)

Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ (\$K)

Bilindiği üzere GAMS terminolojisinde,

İndisler (indices) **sets** olarak

Verilen bilgiler, yani problem girdileri, **parameters** olarak

Karar değişkenleri, **variables** olarak

Kısıtlar ve amaç fonksiyonu da **equations** olarak değerlendirilir.

Problemin parametrelerini oluşturacak olan, şehirler arası uzaklıklar, aşağıdaki tabloda verilmiştir.

<i>Plants</i>	<i>Shipping Distances to Markets (1000 miles)</i>			<i>Supplies</i>
	New York	Chicago	Topeka	
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
<i>Demands</i>	325	300	275	

```

Sets
  i   canning plants   / seattle, san-diego /
  j   markets           / new-york, chicago, topeka / ;

Parameters
  a(i)  capacity of plant i in cases
        /    seattle    350
          san-diego    600  /

  b(j)  demand at market j in cases
        /    new-york    325
          chicago      300
          topeka       275  / ;

Table d(i,j)  distance in thousands of miles
           new-york    chicago    topeka
seattle      2.5        1.7        1.8
san-diego     2.5        1.8        1.4  ;

Scalar f  freight in dollars per case per thousand miles  /90/ ;

Parameter c(i,j)  transport cost in thousands of dollars per case ;

          c(i,j) = f * d(i,j) / 1000 ;

Variables
  x(i,j)  shipment quantities in cases
  z       total transportation costs in thousands of dollars ;

```

```

Positive Variable x ;

```

```

Equations

```

```

    cost          define objective function
    supply(i)     observe supply limit at plant i
    demand(j)     satisfy demand at market j ;

```

```

cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) ;

```

```

supply(i) ..     sum(j, x(i,j))  =l=  a(i) ;

```

```

demand(j) ..     sum(i, x(i,j))  =g=  b(j) ;

```

```

Model transport /all/ ;

```

```

Solve transport using lp minimizing z ;

```

```

Display x.l, x.m ;

```

Problem yukarıda verildiği gibi modellenir ve çözülürse aşağıdaki sonuç elde edilecektir:

	new-york	chicago	topeka
seattle	50.000	300.000	
san-diego	275.000		275.000

Burada görüldüğü üzere, optimum çözümde Seattle şehrinden Topeka'ya ürün gönderilmemektedir.

Fakat, Seattle'dan Topeka'ya bir birim ürün gönderilmek isterse, bunun amaç fonksiyona yapacağı etkinin 0.036 (\$K) veya \$36 olacağı, aşağıdaki marjinal maliyetlerden görülebilir.

	chicago	topeka
seattle		0.036
san-diego	0.009	

Örneğimizde, indisler sets olarak aşağıdaki gibi tanımlanmıştır:

```
Sets
  i  canning plants  / seattle, san-diego /
  j  markets          / new-york, chicago, topeka / ;
```

Bu örnekte setleri birleştirmeden de kullanabilirdik:

```
Set   i  canning plants  / seattle, san-diego / ;
Set   j  markets          / new-york, chicago, topeka / ;
```

Set tanımlanırken * kullanılırsa, ilgili set aritmetik olarak ardışık artan şekilde tanımlanır:

```
Set   t  time periods    /1991*2000/;
Set   m  machines         /mach1*mach24/;
```

Bu tanımlama ile aşağıdaki gibi iki set elde edilebilir.

$$t = \{1991, 1992, 1993, \dots, 2000\}$$
$$m = \{mach_1, mach_2, \dots, mach_{24}\}.$$

Alias

Bir değişken veya parametre, bir kümenin kendisiyle kartezyen çarpımından oluşan kümeyle ilişkili ise, aynı kümeye ait iki farklı indis kullanabilmek için “Alias” özelliğini kullanabiliriz.

```
sets  
i şehirler /1*20/  
;  
Alias(i,j,k);  
Positive variable x(i,j)
```

Not:

Açıklama yazmak için açıklamanın başına * işareti koyun. GAMS bu işaretin devamındaki hiçbir şeyi dikkate almayacaktır.

GAMS'te üç farklı formatta veri girmek mümkündür:

- Liste ile
- Tablo ile
- Doğrudan atama ile

Ayrıca sabit değerler scalar olarak girilir.

Liste ile veri girişi örnekleri:

Parameters

```
a(i)  capacity of plant i in cases
/      seattle      350
      san-diego     600  /
```

```
b(j)  demand at market j in cases
/      new-york     325
      chicago       300
      topeka        275  / ;
```

```
Parameters a(i)  capacity of plant i in cases
/ seattle      350
  san-diego    600  / ;
```

```
Parameters b(j)  demand at market j in cases
/ new-york      325
  chicago       300
  topeka        275  / ;
```

Tablo Kullanımı:

GAMS tablolar kullanarak da kolayca veri girişini mümkün kılar. Örneğin:

```
Table d(i,j)  distance in thousands of miles
              new-york    chicago    topeka
seattle       2.5         1.7         1.8
san-diego     2.5         1.8         1.4  ;
```

Burada, i ve j nin kartezyen çarpımlarından oluşan d matrisi için değerler tanımlanmaktadır.

Doğrudan Atama:

```
Parameter c(i,j)  transport cost in thousands of dollars per case ;  
                c(i,j) = f * d(i,j) / 1000 ;
```

Burada, ilk satırın sonundaki noktalı virgül kullanımına dikkat edilmelidir.

Burada, c matrisindeki bazı öğelerin değeri de doğrudan atama yoluyla belirlenebilir.

Örneğin;

```
c('Seattle','New-York') = 0.40;
```

GAMS'te kullanılabilecek bazı doğrudan atama örnekleri:

```
csquared      = sqr(c);  
e             = m*csquared;  
w            = 1/lamda;  
eoq(i)       = sqrt( 2*demand(i)*ordcost(i)/holdcost(i));  
t(i)         = min(p(i), q(i)/r(i), log(s(i)));  
euclidean(i,j) = qrt(sqr(xi(i) - xi(j) + sqr(x2(i) - x2(j)));  
present(j)    = future(j)*exp(-interest*time(j));
```

Variables

Bilindiği üzere GAMS'te kullanılacak karar değişkenleri önceden tanımlanmalıdır.

```
Variables  
    x(i,j)  shipment quantities in cases  
    z       total transportation costs in thousands of dollars ;
```

Not: Amaç fonksiyonu değerini tutacak olan değişken (z) de tanımlanmalıdır. Burada z için bir domain (indis belirteci) bulunmamaktadır, çünkü z scalar bir değerdir.

Tanımlandığı zaman her değişkenin bir tipi olmalıdır. GAMS'te izin verilen değişken tipleri aşağıdaki gibidir:

<i>Variable Type</i>	<i>Allowed Range of Variable</i>
free(default)	$-\infty$ to $+\infty$
positive	0 to $+\infty$
negative	$-\infty$ to 0
binary	0 or 1
integer	0, 1, ..., 100 (default)

```
Positive variable x ;
```

Equations

GAMS'te denklemler farklı bölümlerde deklare edilir ve tanımlanırlar.

```
Equations
    cost          define objective function
    supply(i)     observe supply limit at plant i
    demand(j)     satisfy demand at market j ;
```

Burada Equation kelimesi hem eşitlik, hem de eşitsizlik durumları için kullanılır.

Toplama Formülü:

Denklemler konusunda daha detaya nmeden, toplama formülü anlatılmalıdır. GAMS'te toplama formülü **Sum** belirteci ile ifade edilir. Taşıma probleminden bir örnek vermek gerekirse:

```
Sum(j, x(i,j))
```

Bu komut, $\sum_j x_{ij}$ 'ye karşılık gelmektedir.



Benzer şekilde;

`Sum((i,j), c(i,j)*x(i,j))` formülü

aşağıdaki ifadeye karşılık gelmektedir:

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

Bu formül aşağıdaki gibi iç içe şekilde de yazılabilirdi:

`Sum(i, Sum(j, c(i,j)*x(i,j)))`



Çarpım Formülü:

GAMS'te çarpım formülü prod komutu ile gerçekleştirilir. Örneğin;

```
prod(j, x(i, j))
```

Komutu aşağıdaki formüle karşılık gelmektedir:

$$\prod_j x_{ij}$$

NOT: Toplam ve çarpım komutları, scalar değerleri girerken de kullanılabilir. Örneğin:

```
scalar tots supply    total supply over all plants;  
tots supply = sum(i, a(i));
```

Π

Equation Tanımlamaları:

Eşitlik/eşitsizlik tanımlamaları GAMS'te en karmaşık kısımdır. Çünkü fazla sayıda denklem türü bulunmaktadır. Ulaştırma problem örneğindeki denklemler aşağıdaki şekilde tanımlanabilir:

```
cost ..      z  =e=  sum((i,j), c(i,j)*x(i,j)) ;

supply(i) ..  sum(j, x(i,j))  =l=  a(i) ;

demand(j) ..  sum(i, x(i,j))  =g=  b(j) ;
```

Burada set kullanımından kaynaklanan büyük kolaylık vardır. Örneğin demand(j) kısıtı, j domain için aşağıdaki açık formda yazılan denklemlere karşılık gelmektedir:

```
DEMAND(new-york)..X(seattle,new-york) + X(san-diego,new-york)=G=325 ;
DEMAND(chicago).. X(seattle,chicago) + X(san-diego,chicago)  =G=300 ;
DEMAND(topeka)..  X(seattle,topeka) + X(san-diego,topeka)      =G=275 ;
```

Dolayısıyla, “set” kullanımı özellikle fazla sayıda değişkenden oluşan gerçek hayat problemlerinde büyük kolaylık sağlamaktadır.

İlişkisel operatorler daha önce de ifade edildiği gibi aşağıda verilmektedir:

=l=	less than or equal to
=g=	greater than or equal to
=e=	equal to

Burada dikkat edilmesi gereken husus “=” ile “=e=” kullanımı arasındaki farktır.

“=” operatörü doğrudan atamalarda kullanılır. Örneğin bir scalar ifadenin değerini atamak için kullanılır.

Fakat “=e=” ifadesi denklemlerin tanımlamasında eşittir anlamında kullanılır.

NOT: Denklemler model içerisinde herhangi bir yerde tanımlanabilir. Fakat kullanılacak tüm değişkenler daha önceden tanımlanmış olmalıdır.

Denklemler GAMS tarafından otomatik olarak en sade şekline dönüştürülürler.

Amaç Fonksiyonu (Objective Function)

GAMS'te objective olarak tanımlanan ayrıca bir ifade yoktur.

Bir amaç tanımlamak için, öncelikle scalar değerli ve free-form'da bir değişken tanımlanmalıdır.

Amaç fonksiyonu değerini tutacak değişken indis (domain) içeremez.

Ulaştırma probleminden örnek verecek olursak:

```
Variables  
      z      total transportation costs in thousands of dollars ;
```

...

```
cost ..      z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
```

Model Komutu

model komutu oldukça kesin bir anlama sahiptir. Çözülecek modeli isimlendirmek ve modelin kapsamını belirlemek için kullanılır.

Örneğin;

```
model transport /all/ ;
```

Komutu ile, **transport** olarak isimlendirilen model, **/all/** belirteci ile tüm denklemleri içerecek şekilde oluşturulur.

Başka bir ifade ile aynı model;

```
model transport / cost, supply, demand / ;
```

komutu ile de oluşturulabilir.

Bu özellik, ileri düzey kullanıcılar için tek çalıştırmada birden fazla modelin çalıştırılabilmesi açısından oldukça faydalıdır.

Model Komutu

► Example:

```
model m / all /;  
model m / objective, e1 /;
```

Attributes:

set by user		set by solver	
iterlim	iteration limit	iterusd	number of iterations
reslim	time limit in seconds	resusd	solving time
optcr	relative gap tolerance	modelstat	model status
optfile	number of solver options file	solvestat	solver status
...		...	

Example:

```
m.reslim = 60;      m.optcr = 0;  
solve m minimize objvar using MINLP;  
display m.resusd;
```

Solve Komutu

Model oluşturulduktan sonra `solve` komutu çağrılarak çalıştırılır. Örneğimizdeki kullanım şekli:

```
solve transport using lp minimizing z ;
```

`solve` komutunu model ismi izler ve `using` anahtar kelimesinden sonra model türü belirtilir.

Ardından çözücünün yönü `maximizing` veya `minimizing` belirteçleri ile belirtildikten sonra en iyilenecek değişkenin ismi yazılır (z).

<code>lp</code>	for linear programming
<code>qcp</code>	for quadratic constraint programming
<code>nlp</code>	for nonlinear programming
<code>dnlp</code>	for nonlinear programming with discontinuous derivatives
<code>mip</code>	for mixed integer programming
<code>rmip</code>	for relaxed mixed integer programming
<code>miqcp</code>	for mixed integer quadratic constraint programming

rmiqcp	for relaxed mixed integer quadratic constraint programming
minlp	for mixed integer nonlinear programming
rminlp	for relaxed mixed integer nonlinear programming
mcp	for mixed complementarity problems
mpec	for mathematical programs with equilibrium constraints
rmpec	for relaxed mathematical program with equilibrium constraints
cns	for constrained nonlinear systems
emp	for extended mathematical programming

Display Komutu

Solve komutu ile model çalıştırıldıktan sonra elde edilen sonuçlar rapor dosyasında gösterilmektedir. Bunun yanında bazı değişkenler için elde edilen sonuçların ayrıca (özel olarak) gösterilmesi istenebilir. Örneğin;

```
display x.l, x.m ;
```

komutu $x(i,j)$ değişkeninin optimum sonuç için elde edilen değerlerini ve $x(i,j)$ için indirgenmiş maliyetlerini rapor edecektir.

.lo, .l, .up, .m veritabanı:

.lo	lower bound
.l	level or primal value
.up	upper bound
.m	marginal or dual value

Ulaştırma Problemine Ait Tam Model

```
Sets
i canning plants / seattle, san-diego /
j markets          / new-york, chicago, topeka / ;
```

Parameters

```
a(i) capacity of plant i in cases
    / seattle   350
      san-diego 600 /

b(j) demand at market j in cases
    / new-york 325
      chicago  300
      topeka   275 / ;
```

Table d(i,j) distance in thousands of miles

	new-york	chicago	topeka
seattle	2.5	1.7	1.8
san-diego	2.5	1.8	1.4

Scalar f freight in dollars per case per thousand miles /90/ ;

```
Parameter c(i,j) transport cost in thousands of dollars per case ;
c(i,j) = f * d(i,j) / 1000 ;
```

Parameter c(i,j) transport cost in thousands of dollars per case ;
 $c(i,j) = f * d(i,j) / 1000$;

Variables

```
x(i,j) shipment quantities in cases
z      total transportation costs in thousands of dollars ;
```

Positive Variable x;

Equations

```
cost      define objective function
supply(i) observe supply limit at plant i
demand(j) satisfy demand at market j ;
```

```
cost ..      z =e= sum((i,j), c(i,j)*x(i,j)) ;
supply(i) .. sum(j, x(i,j)) =l= a(i) ;
demand(j) .. sum(i, x(i,j)) =g= b(j) ;
```

Model transport /all/ ;

Solve transport using lp minimizing z ;

Display x.l, x.m ;

S O L V E S U M M A R Y

MODEL transport OBJECTIVE z
 TYPE LP DIRECTION MINIMIZE
 SOLVER CPLEX FROM LINE 42

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 153.6750

RESOURCE USAGE, LIMIT 0.003 1000.000
 ITERATION COUNT, LIMIT 4 2000000000

IBM ILOG CPLEX Jul 4, 2010 23.5.1 WIN 18414.18495 VS8 x86/MS Windows
 Cplex 12.2.0.0, GAMS Link 34

LP status(1): optimal
 Optimal solution found.
 Objective : 153.675000

---- 44 VARIABLE x.L shipment quantities in cases

	new-york	chicago	topeka
seattle	50.000	300.000	
san-diego	275.000		275.000

---- 44 VARIABLE x.M shipment quantities in cases

	chicago	topeka
seattle		0.036
san-diego	0.009	

Model Status

Model status: what the solution looks like

1	Optimal (global optimality)
2, 8	Locally Optimal
3, 18	Unbounded
4, 10, 19	Infeasible (proven to be infeasible)
5	Locally Infeasible
6	Intermediate Infeasible
7	Intermediate Nonoptimal (feasible for continuous models)
9	Intermediate Non-integer
11	License Problem
12, 13	Error
14	No Solution (but expected so)
15, 16, 17	Solved for CNS models

Solver Status

Solver status: state of the solver program

1	Normal Completion
2	Iteration Interrupt
3	Resource Interrupt (timelimit reached)
4	Terminated by Solver
5	Evaluation Error Limit
6	Capability Problems
7	Licensing Problems
8	User Interrupt
9, ..., 13	setup, solver, system errors

Option Komutu

- ▶ some important parameters:

keyword	meaning	default
iterlim	iteration limit	2000000000
reslim	time limit	1000 (!!)
optca	absolute gap tolerance	0.0
optcr	relative gap tolerance	0.1 (!!)
LP	choice of LP solver	CPLEX
NLP	choice of NLP solver	CONOPT
...		

- ▶ Example:

```
option iterlim = 100, optcr = 0;  
solve icesale using mip min kosten;  
option mip = cbc;  
solve icesale using mip min kosten;
```

ÖDEVLER

ATK Ulastirma Problemi

ATK- Beyaz dört müşterisinin taleplerini karşılamak için üç potansiyel fabrikayı değerlendirmektedir. Her hangi bir fabrikayı açarsa firma 200000TL'lik yatırım yapması gerekmektedir.

- Bir fabrikada üretilip, bir müşteriye gönderme birim maliyetleri tabloda verilmiştir.
- Firmanın talepleri en düşük maliyetle karşılaması için gerekli TP modelini kurunuz.

	Müşteri 1	Müşteri 2	Müşteri 3	Müşteri 4	Kapasite
Fabrika 1	8TL	6TL	10TL	9TL	60,000
Fabrika 2	9TL	12TL	13TL	7TL	55,000
Fabrika 3	14TL	9TL	16TL	5TL	45,000
Talepler	40,000	30,000	25,000	15,000	

Matematiksel Model

► Karar değişkenleri

► x_{ij} : i fabrikasından j müşterisine gönderilen miktar ($i = 1,2,3; j = 1,2,3,4$)

► y_i : i fabrikasının açılması ($i = 1,2,3$)

$$y_i = \begin{cases} 1 & i \text{ fabrikası açılırsa} \\ 0 & \text{açılmazsa} \end{cases}$$

► TP modeli

$$\text{Min } \sum_i \sum_j c_{ij} x_{ij} + 200000 \sum_i y_i$$

Öyle ki; $\sum_j x_{ij} \leq k_i y_i \quad (i=1,2,\dots,m)$ kapasite kısıtları

$$\sum_i x_{ij} \geq t_j \quad (j=1,2,\dots,n) \quad \text{Talep kısıtları}$$

$$x_{ij} \geq 0, y_i \in \{0, 1\}$$

```
sets i fabrikalar /1, 2, 3/
      j müşteriler /1*4/ ;
```

```
table c(i,j)
      1   2   3   4
1     8   6  10   9
2     9  12  13   7
3    14   9  16   5 ;
```

```
Scalar fa fabrika acma maliyeti /200000/;
```

```
parameter talep(j) /1 40000, 2 30000, 3 25000, 4 15000 /;
```

```
parameter kapasite(i) / 1 60000, 2 55000, 3 45000 /;
```

```
positive variables x(i,j);
```

```
binary variables y(i);
```

```
variable z;
```

```
equations amac, kapasitekisiti, talepkisiti ;
```

```
    amac..                z =e= sum((i,j), x(i,j)*c(i,j)) + fa*sum(i, y(i));
```

```
    kapasitekisiti(i).. sum(j, x(i,j)) =l= kapasite(i)*y(i);
```

```
    talepkisiti(j)..      sum(i, x(i,j)) =g= talep(j);
```

```
model ulastirma /all/;
```

```
solve ulastirma using MIP minimizing z;
```

```
display x.l, y.l;
```

S O L V E S U M M A R Y

MODEL ulastirma OBJECTIVE z
TYPE MIP DIRECTION MINIMIZE
SOLVER CPLEX FROM LINE 22

**** SOLVER STATUS 1 Normal Completion
**** MODEL STATUS 8 Integer Solution
**** OBJECTIVE VALUE 1290000.0000

RESOURCE USAGE, LIMIT 0.237 1000.000
ITERATION COUNT, LIMIT 13 20000000000

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	5000.000	+INF	.
1.2	.	30000.000	+INF	.
1.3	.	25000.000	+INF	.
1.4	.	.	+INF	3.000
2.1	.	35000.000	+INF	.
2.2	.	.	+INF	5.000
2.3	.	.	+INF	2.000
2.4	.	15000.000	+INF	.
3.1	.	.	+INF	7.000
3.2	.	.	+INF	4.000
3.3	.	.	+INF	7.000
3.4	.	.	+INF	.

---- VAR y

	LOWER	LEVEL	UPPER	MARGINAL
1	.	1.000	1.000	1.4000E+5
2	.	1.000	1.000	2.0000E+5
3	.	.	1.000	1.1000E+5

	LOWER	LEVEL	UPPER	MARGINAL
--	-------	-------	-------	----------

---- VAR z -INF 1.2900E+6 +INF .

Elektronik Ürün Üretimi Problemi

Firmanız, LCD TV, Muzik Calar ve Hoparlor uretimi yapmaktadır. Bu urunlerin uretiminde, Govde, LCD Ekran, Hoparlor, Guc unitesi ve Cesitli Elektronik Parcalari "ortak" olarak kullanılmaktadır. Kullanilacak parcalarin tedarigi sınırlı iken karı maksimize edecek optimum uretim miktarlarini bulunuz.

Part Requirements by Product				Malzeme Stok Miktarlari
Parca Adi	LCD TV	Muzik Calar	Hoparlor	Mevcut
Govde	1	1	0	450
LCDEkran	1	0	0	250
Hoparlor	2	2	1	800
Guc Unitesi	1	1	0	450
Elektronik Parca	2	1	1	600
Birim Kar	\$75	\$50	\$35	

```

sets
i /1*3/

variables
x(i)
z;

positive variables
x(i);

parameters
Govde(i) Govde
/ 1 1
  2 1
  3 0/

L(i) LCD
/ 1 1
  2 0
  3 0/

H(i) Hoparlör
/ 1 2
  2 2
  3 1/

G(i) güç
/ 1 1
  2 1
  3 0/

E(i) elektronik
/ 1 2
  2 1
  3 1/

```

```

p(i) satış fiyat
/ 1 75
  2 50
  3 35/
;

scalars
GovdeK govde stoğu /450/
LK LCD stoğu /250/
HK hoparlör stoğu /800/
GK güç ünitesi /450/
EK elektronik stoğu /600/
;

equations
amac
kisit1
kisit2
kisit3
kisit4
kisit5
;

amac.. z =e= sum(i, p(i)*x(i));
kisit1.. sum(i, govde(i)*x(i)) =l= GovdeK;
kisit2.. sum(i, L(i)*x(i)) =l= LK;
kisit3.. sum(i, H(i)*x(i)) =l= HK;
kisit4.. sum(i, G(i)*x(i)) =l= GK;
kisit5.. sum(i, E(i)*x(i)) =l= EK;

model ornek5 /all/;
solve ornek5 using lp maximizing z;

```

S O L V E S U M M A R Y

MODEL ornek5 OBJECTIVE z
 TYPE LP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 68

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 25000.0000

RESOURCE USAGE, LIMIT 0.003 1000.000
 ITERATION COUNT, LIMIT 3 20000000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	400.000	450.000	.
---- EQU kisit2	-INF	200.000	250.000	.
---- EQU kisit3	-INF	800.000	800.000	12.500
---- EQU kisit4	-INF	400.000	450.000	.
---- EQU kisit5	-INF	600.000	600.000	25.000

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1	.	200.000	+INF	.
2	.	200.000	+INF	.
3	.	.	+INF	-2.500

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR z	-INF	25000.000	+INF	.

Toplama Formülü Kuralları

5

Kaynak: Dhazn Gillig and Bruce A. McCarl, Introduction to GAMS: Summation Notation with GAMS, Department of Agricultural Economics, Texas A&M University (*from Appendix A in McCarl and Spreen's text book on* <http://ageco.tamu.edu/faculty/mccarl/mccspr/newa1.pdf>)

GAMS'te toplama formülü kullanırken indis kullanımı kurallarına dikkat edilmelidir. Aşağıda bununla ilgili bazı detaylı örnek ve açıklamalar verilmiştir.

Öncelikle, örneklerimizde kullanmak üzere bazı değişkenler tanımlayalım.

$$\begin{array}{lll} x_1 = 1 & y_{11} = 2 & y_{12} = 3 \\ x_2 = 2 & y_{21} = 4 & y_{22} = 1 \\ x_3 = 3 & y_{31} = 1 & y_{32} = 4 \end{array}$$

Şimdi bazı toplam formüllerinden örnekler verelim.

1) Bir değişkenin aldığı değerlerin toplamı matematiksel olarak aşağıdaki gibi ifade edilir:

$$\sum_i x_i \qquad \sum_{i=1}^3 x_i = x_1 + x_2 + x_3 = 1 + 2 + 3 = 6$$

Bu matematiksel ifadenin GAMS'teki karşılığı: `Toplam1 = sum(i, x(i));`

2) İç içe toplam formülünün kullanımı (iki indisli toplama formülü) matematiksel olarak aşağıdaki gibidir:

$$\sum_i \sum_j y_{ij} = y_{11} + y_{12} + y_{21} + y_{22} + y_{31} + y_{32} = 15$$

Bu ifadenin GAMS'te kodlanması şu şekildedir: `Toplam2 = sum((i,j), y(i,j));`

3) Aynı indeksi paylaşan iki toplam formülünün kullanımı matematiksel olarak aşağıdaki gibidir:

$$\sum_i \left(x_i + \sum_j y_{ij} \right) = \sum_i x_i + \sum_i \sum_j y_{ij} = x_1 + y_{11} + y_{12} + x_2 + y_{21} + y_{22} + x_3 + y_{31} + y_{32} = 21$$

Bu ifadenin GAMS'te kodlanması şu şekildedir:

`Toplam3 = sum(i, x(i) + sum(j, y(i,j)));`

veya `Toplam3 = sum(i, x(i)) + sum((i,j), y(i,j));`

Diğer yandan, tek bir i indisi için tüm y_{ij} değerlerini toplayacak olsaydık, örneğin;

$$A_i = x_i + \sum_j y_{ij} = x_i + y_{i1} + y_{i2}$$

ifadesi GAMS'te şu şekilde kodlanır: $A(i) = x(i) + \text{sum}(j, y(i, j)) ;$

Buradan hareketle, örneğin, bir *fabrikanın sevk edeceği ürün toplamının fabrika kapasitesini aşmaması için yazılacak olan ulaştırma problem kısıtı*

$$\sum_j x_{ij} \leq a_i \quad \forall i$$

GAMS'te şu şekilde ifade edilebilir: $\text{supply}(i) = \text{sum}(j, x(i, j)) = a(i) ;$

Ortak indisli toplamlar birleştirilebilir veya ayrılabilir:

$$\sum_i (x_i + z_i) = \sum_i x_i + \sum_i z_i$$

Bu ifadenin GAMS'teki karşılığı: `Toplam4 = sum(i, x(i)+z(i));`

veya `Toplam4 = sum(i, x(i)) + sum(i, z(i));`

Doğru Denklem Yazımı

Doğru denklem yazımı için uyulması gereken temel kurallar vardır.

1) Tekli denklem yazımı

Kullanılan tüm indisler toplama formülünün içinde tanımlanmış olmalıdır.

Aşağıdaki gibi bir denklem doğrudur:

$$B = \sum_i \sum_j \sum_k p_{ijk} + \sum_m \sum_n q_{mn}$$

Bu ifadenin GAMS'teki karşılığı:

```
EQB.. B =e= sum((i,j,k), p(i,j,k)) + sum((m,n), q(m,n));
```

Fakat aşağıdaki matematiksel ifadeler YANLIŞTIR:

$$C = p_{ijk} + q_{mn}$$

$$D = \sum_i \sum_j p_{ijk} + \sum_m \sum_n q_{mn}$$

Yine GAMS'teki aşağıdaki ifadeler de YANLIŞTIR :

$$\text{EQC} \dots C = e = p(i, j, k) + q(m, n) ;$$

$$\text{EQD} \dots D = e = \text{sum}((i, j), p(i, j, k)) + \text{sum}((m, n), q(m, n)) ;$$

Bu şekilde bir yazım aşağıdaki hatayla sonuçlanacaktır:

..error message \$149 "UNCONTROLLED SET ENTERED AS CONSTANT"

2) Denklem Grubu Yazımı

Kural 1: Denklem grubu indisleri tanımlanmış olmalıdır.

Örneğin, C_i 'nin tüm i indisleri için değerinin 2 olması istenirse:

$$C_i = 2 \quad \forall i \quad \text{veya} \quad C_i = 2 \quad i = 1, 2, \dots, n$$

GAMS'teki yazım: `denkc (i) .. C (i)=2 ;`

Diğer örnekler:

$$D_{ij} = 2 \quad \forall i, j \quad \text{denkd (i, j) .. D (i, j)=2 ;}$$

$$E_{ij} = F_i \quad \forall i, j \quad \text{denke (i, j) .. E (i, j)=F (i) ;}$$

Kural 2: “Her bir ($\forall$)” indis elemanı için tanımlanacak denklemlerde, “her bir (for all veya $\forall$)” ifadesi içinde tanımlanmamış tüm indisler toplam formülü içinde verilmelidir.

Aşağıdaki yazımlar doğrudur:

$$\sum_k p_{ijk} = h_i \quad \forall i, j$$

$$\text{eq1}(i, j) \dots \text{sum}(k, p(i, j, k)) = h(i);$$

$$\sum_j \sum_k p_{ijk} = g_i \quad \forall i$$

$$\text{eq2}(i) \dots \text{sum}((j, k), p(i, j, k)) = g(i);$$

$$p_{ijk} = g_i \quad \forall i, j, k$$

$$\text{eq3}(i, j, k) \dots p(i, j, k) = g(i);$$

Fakat aşağıdaki şekilde yazım yanlıştır:

$$\sum_k p_{ijk} = h_i \quad \forall i \quad \text{eq4 (i) .. sum(k, p(i,j,k)) =e= g(i) ;}$$

Bu şekilde bir yazım da aşağıdaki hatayla sonuçlanacaktır:

..error message \$149 “UNCONTROLLED SET ENTERED AS CONSTANT”

Kural 3: “Her bir ($\forall$)” indis elemanı içinde tanımlanacak bir indisten, tüm terimler içinde en az bir tane bulunmalıdır.

Aşağıdaki yazımlar doğrudur:

$$\sum_j \sum_k r_{ijkm} + \sum_j s_{ij} = N_{im} \quad \forall i, m$$

`eqN(i,m) .. sum((j,k), r(i,j,k,m)) + sum(j, s(i,j)) = N(i,m);`

Fakat aşağıdaki şekilde yazım yanlıştır:

$$p_{ijk} = L2 \quad \forall i \quad \text{eqL(i) .. p(i,j,k) = L2;}$$

Yine aşağıdaki hatayla sonuçlanacaktır:

..error message \$149 “UNCONTROLLED SET ENTERED AS CONSTANT”

Kural 4: Eğer bir indis “her bir ($\forall$)” ifadesi içinde tanımlanmışsa, bu indis aynı zamanda toplam formülü içinde de tanımlanamaz.

Aşağıdaki şekilde yazımlar kesin olarak yanlıştır:

$$\sum_i W_i = N_i \quad \forall i \quad \text{eq} W(i) \dots \text{sum}(i, W(i)) = N(i) ;$$

$$W_i = \sum_i N_i \quad \forall i \quad \text{eq} N(i) \dots W(i) = \text{sum}(i, N(i)) ;$$

Bu durumda şu hatayla karşılaşılacaktır: ...error message \$125 “Set is under control already.”

GAMS Modelleme Örnekleri ve Çalışma Soruları

6

Kaynak: Dhazn Gillig and Bruce A. McCarl, Introduction to GAMS: Summation Notation with GAMS, Department of Agricultural Economics, Texas A&M University (*from Appendix A in McCarl and Spreen's text book on* <http://ageco.tamu.edu/faculty/mccarl/mccspr/newa1.pdf>)

Petrol Karışım Problemi

Sunco, oktan dereceleri ve sülfür oranları farklı üç tip ham petrolün (H1, H2, H3) karıştırılması ile üç tip benzin (B1, B2, B3) üretmektedir. Benzinlerin oktan dereceleri ve sülfür oranları belli standartları sağlamalıdır:

- B1 için ortalama oktan derecesi en az 10, sülfür oranı en fazla %2 olmalıdır
- B2 için ortalama oktan derecesi en az 8, sülfür oranı en fazla %4 olmalıdır
- B3 için ortalama oktan derecesi en az 6, sülfür oranı en fazla %3 olmalıdır

Firmanın her benzin tipi için en fazla satabileceği talepler B1, B2 ve B3 için sırasıyla 3000, 2000 ve 1000 varildir. Bununla birlikte firma reklam yaparak talebini arttırabilmektedir. Herhangi bir benzinde 1 dolarlık reklam, talebi 10 varil arttırmaktadır. Hammaddelerin oktan dereceleri, sülfür oranları ve alış fiyatları ile benzinlerin satış fiyatları aşağıda verilen tablolardaki gibi ise Sunco'nun karını enbüyükleyecek çözümü bulunuz.

Ham petrol	Oktan	Sülfür (%)	Alış fiyatı (\$/varil)	Benzin	Satış fiyatı (\$/varil)
1	12	1	45	1	70
2	6	3	35	2	60
3	8	5	25	3	50

Karar değişkenleri

x_{ij} : i . hammaddeden j . benzine konulan miktar (varil), $i = 1,2,3$; $j=1,2,3$.

r_j : j . benzin için yapılan reklam (\$), $j=1,2,3$.

Amaç fonksiyonu (karı enbüyüklemek)

Maks Z = Kar = gelir – maliyet

$$\text{Maks } Z = (70 \sum_i x_{i1} + 60 \sum_i x_{i2} + 50 \sum_i x_{i3}) - (45 \sum_j x_{1j} + 35 \sum_j x_{2j} + 25 \sum_j x_{3j}) - \sum_j r_j$$

Kısıtlar

Oktan derecesi

$$12x_{11} + 6x_{21} + 8x_{31} \geq 10 (x_{11} + x_{21} + x_{31})$$

benzin 1 oktan derecesi

$$12x_{12} + 6x_{22} + 8x_{32} \geq 8(x_{12} + x_{22} + x_{32})$$

benzin 2 oktan derecesi

$$12x_{13} + 6x_{23} + 8x_{33} \geq 6 (x_{13} + x_{23} + x_{33})$$

benzin 3 oktan derecesi

Sülfür oranları

$$(.01x_{11} + .03x_{21} + .05x_{31})/(x_{11} + x_{21} + x_{31}) \leq .02 \rightarrow$$

$$x_{11} + 3x_{21} + 5x_{31} \leq 2 (x_{11} + x_{21} + x_{31})$$

benzin 1 sülfür oranı

$$x_{12} + 3x_{22} + 5x_{32} \leq 4 (x_{12} + x_{22} + x_{32})$$

benzin 2 sülfür oranı

$$x_{13} + 3x_{23} + 5x_{33} \leq 2 (x_{13} + x_{23} + x_{33})$$

benzin 3 sülfür oranı

Talepler

$$\sum_i x_{ij} \leq T_j + 10r_j \quad \forall j. \quad (T_j: j. benzinin reklamsız talebi)$$

İşaret sınırlamaları

EMM3208 Optimizasyon Teknikleri - Doç. Dr. İbrahim Küçükkoç

$$x_{ij}, r_j \geq 0, \forall i,j.$$

```

sets
  i hammadde /1*3/
  j benzin /1*3/;

parameter
  demand(j) j türü benzine olan talep
  /1 3000
  2 2000
  3 1000/;

parameter
  hmoktan(i) i türü hammaddeye ait oktan derecesi
  /1 12
  2 6
  3 8/;

parameter
  hmsulfur(i) i türü hammaddenin sülfür oranı
  /1 1
  2 3
  3 5/;

parameter
  hmalis(i) i türü hammaddenin alış fiyatı
  /1 45
  2 35
  3 25/;

parameter
  bnoktan(j) j türü benzinin oktan derecesi
  /1 10
  2 8
  3 6/;

```

EMM3208 Optimizasyon Teknikleri - Doç. Dr. İbrahim Küçükkoç

```

parameter
  bnsulfur(j) j türü benzinin sülfür oranı
  /1 2
  2 4
  3 3/;

parameter
  bnsatis(j) j türü benzinin satış fiyatı
  /1 70
  2 60
  3 50/;

positive variable x(i,j);
*i. hammaddeden j. benzine konulan miktar (varıl)
positive variable r(j);
*j türü benzin için yapılan reklam
variable z;
*toplam kar
scalar f /10/;
*1 dolarlık reklam, talebi 10 varıl arttırır demek için kullanılan sabit

equations
  kar amaç fonksiyonu
  supply(j) talebin karşılanmasını sağlamak üzere kullanılan kısıt
  oktan(j) oktan kısıtı
  sulfur(j) sulfur kısıtı;

kar.. z=e=sum((i,j), bnsatis(j)*x(i,j))-sum((i,j), ((hmalis(i)*x(i,j)-r(j))));
supply(j).. sum((i), (x(i,j))) =l= (demand(j)+(f*r(j)));
oktan(j).. sum((i), (bnoktan(j)*x(i,j))) =l= sum((i), (hmoktan(i)*x(i,j)));
sulfur(j).. sum((i), (bnsulfur(j)*x(i,j))) =l= sum((i), (hmsulfur(i)*x(i,j)));

model ornek6 /all/;
solve ornek6 using lp maximizing z;

```

S O L V E S U M M A R Y

MODEL ornek6 OBJECTIVE z
 TYPE LP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 71

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 3 Unbounded
 **** OBJECTIVE VALUE 60000.0000

RESOURCE USAGE, LIMIT 0.033 1000.000
 ITERATION COUNT, LIMIT 3 20000000000

IBM ILOG CPLEX Jul 4, 2010 23.5.1 WIN 18414.18495 VS8 x86/MS
 Cplex 12.2.0.0, GAMS Link 34

LP status(4): unbounded or infeasible
 Presolve found the problem infeasible or unbounded.
 Rerunning with presolve turned off.

LP status(2): unbounded
 Dual infeasible or unbounded. Switching to primal to aid diagno
 LP status(2): unbounded
 Model has an unbounded ray.

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL	
1.1	.	.	+INF	.	
1.2	.	250.000	+INF	.	
1.3	.	.	+INF	5.000	NOPT
2.1	.	.	+INF	.	
2.2	.	500.000	+INF	.	
2.3	.	.	+INF	15.000	NOPT
3.1	.	.	+INF	240.000	NOPT
3.2	.	1250.000	+INF	.	
3.3	.	.	+INF	25.000	NOPT

---- VAR r

	LOWER	LEVEL	UPPER	MARGINAL	
1	.	.	+INF	3.000	NOPT
2	.	.	+INF	303.000	UNBND
3	.	.	+INF	3.000	NOPT

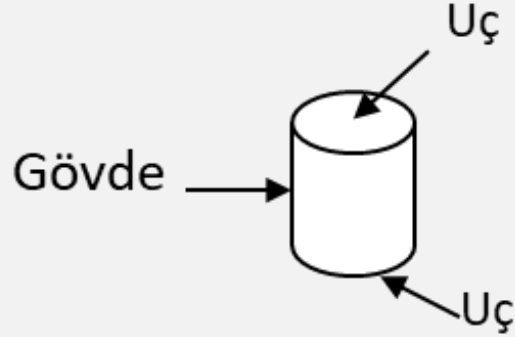
	LOWER	LEVEL	UPPER	MARGINAL
--	-------	-------	-------	----------

---- VAR z

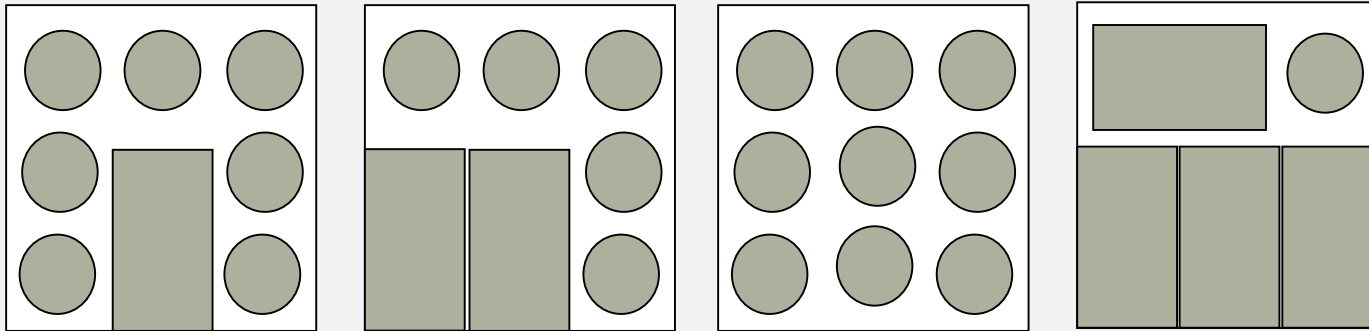
	-INF	60000.000	+INF	.
--	------	-----------	------	---

Plaka Kesim Problemi

Aşağıda gösterilen şekilde gerçekleştirilen bir teneke kutu üretimini düşününüz.



Şekilde de gösterildiği gibi her teneke kutu bir adet gövde ve iki adet uçtan oluşmaktadır. Bu parçaların teneke plakalardan kesimi aşağıda verilen dört ayrı şablona göre yapılmaktadır:



Bu şablonlar için ayrıca aşağıdaki bilgiler verilmiştir:

	Şablon			
	1	2	3	4
Gövde Sayısı	1	2	0	4
Uç sayısı	7	5	9	1
Atık miktarı	12 cm ²	10 cm ²	17 cm ²	6 cm ²
Kesim zamanı (dakika)	3	2.5	4	1.5

Bir adet kutunun satışından 12 TL kazanılmaktadır. Bunun yanı sıra her bir cm² atık 0.5 TL'ye mal olmaktadır. Bir hafta 35 saatten oluşmaktadır ve haftada en fazla 27 adet teneke plaka sağlanabilmektedir.

Hafta başında elde hiç stok bulunmamaktadır. Hafta sonunda kullanılmayan her bir gövde parçası 2 TL, her bir uç parça ise 0.5 TL maliyet oluşturmaktadır. Üretilen her kutu satılabilmektedir.

Bu şartları göz önünde bulundurarak firmanın haftalık karını maksimum yapmak için üreteceği teneke sayısını verecek bir doğrusal programlama modeli oluşturunuz.

xi: Kesilen i numaralı şablon sayısı (i: 1, 2, 3, 4)

y: Üretilen kutu sayısı ise

p: Bir kutunun satış fiyatı 12 TL ($f = 12$) olmak üzere kutuların satışından elde edilen
toplam gelir: $f \cdot y$ 'dir

c: santimetrekare atık maliyeti ($m = 0.5$) ve

ai: i şablonunun kullanılması neticesinde bir plakadan çıkan atık miktarı olmak üzere
toplam atık maliyeti: $c \sum_{i=1}^4 a_i x_i$

hg: bir gövde parçasını elde bulundurma maliyeti (holding cost) ($hg = 2$)

hu: bir uç parçasını elde bulundurma maliyeti (holding cost) ($hu = 0.5$)

gi : i şablonundaki gövde sayısı

ui : i şablonundaki uç sayısı ise ve üretilen her kutu satılabiliyorsa

toplam elde bulundurma maliyeti:

$$hg \sum_{i=1}^4 g_i x_i + hu \sum_{i=1}^4 u_i x_i$$

Bu durumda ***amaç fonksiyonu (toplam kar):***

$$z_{maks} = fy - c \sum_{i=1}^4 a_i x_i - hg \sum_{i=1}^4 g_i x_i - hu \sum_{i=1}^4 u_i x_i$$

Problemde malzeme ve kapasite kısıtı olmak üzere iki kısıt altında optimum karı sağlayacak şekilde hangi şablonun kaç plakada kullanılması gerektiğinin bulunması istenmektedir.

k_i : i şablonunun kesim süresi (dakika) ve

th : haftalık kesim saati kapasitesi (time capacity) ($th = 35$)

1 saat 60 dakikaysa

tc : haftalık kesim dakika kapasitesi = $60 * 35$

oq : haftalık temin edilebilen plaka satışı (order quantity) ($oq = 27$) olmak üzere kısıtlar;

$$\sum_{i=1}^4 k_i x_i \leq tc$$

$$\sum_{i=1}^4 x_i \leq oq$$

Problemde verilen üretim siteminin malzeme ve kapasite kısıtları haricinde iki sistem kısıtı daha bulunmaktadır. Bunlar;

- Ürünün yapısı gereği (ürün ağacına göre) 1 kutuda 1 gövde bulunmaktadır, dolayısıyla üretilen kutu sayısı kesilen şablonlardaki toplam gövde sayısından büyük olamaz.

$$y \leq \sum_{i=1}^4 g_i x_i$$

- Yine ürünün yapısı gereği (ürün ağacına göre) 1 kutuda 2 gövde bulunmaktadır, dolayısıyla üretilen kutu sayısının 2 katı kesilen şablonlardaki toplam gövde sayısından küçüktür.

$$2y \leq \sum_{i=1}^4 u_i x_i$$

Değişkene bağımlı ve değişkenden bağımsız parametreler ile sabitler yandaki şekilde tanımlanır.

```
sets i /1*4/

scalar
d bir saat 60 dakika /60/;

parameters
a(i) atik miktarı /1 12, 2 10, 3 17, 4 6/
g(i) sablonlardaki govde sayisi /1 1, 2 2, 3 0, 4 4/
u(i) sablonlardaki uc sayisi /1 7, 2 5, 3 9, 4 1/
k(i) kesim zamanı /1 3, 2 2.5, 3 4, 4 1.5/
p bir kutunun satis fiyatı /12/
c santimetrekare atik maliyeti /0.5/
hg bir govdenin elde bulundurma maliyeti /2/
hu bir ucun elde bulundurma maliyeti /0.5/
oq haflalık temin edilebilen plaka sayisi /27/
ch haftalık kesim saati kapasitesi /35/
cm haftalık kesim dakikası kapasitesi;

cm=d*ch;
```

İndisli değişkenler ve tamsayılı
değişkenler yandaki şekilde
ifade edilir.

```
sets i /1*4/
```

```
scalar
```

```
d bir saat 60 dakika /60/;
```

```
parameters
```

```
a(i) atik miktarı /1 12, 2 10, 3 17, 4 6/
```

```
g(i) sablonlardaki govde sayısı /1 1, 2 2, 3 0, 4 4/
```

```
u(i) sablonlardaki uc sayısı /1 7, 2 5, 3 9, 4 1/
```

```
k(i) kesim zamanı /1 3, 2 2.5, 3 4, 4 1.5/
```

```
p bir kutunun satış fiyatı /12/
```

```
c santimetrekare atik maliyeti /0.5/
```

```
hg bir govdenin elde bulundurma maliyeti /2/
```

```
hu bir ucun elde bulundurma maliyeti /0.5/
```

```
oq haftalık temin edilebilen plaka sayısı /27/
```

```
ch haftalık kesim saati kapasitesi /35/
```

```
cm haftalık kesim dakika kapasitesi;
```

```
cm=d*ch;
```

```
variables
```

```
y
```

```
x(i)
```

```
z;
```

```
integer variables
```

```
y
```

```
x(i);
```

Belirtilen bir kümedeki (setteki) değişkenlerin toplamı “sum” formülüyle ifade edilir ve ilgilendiğimiz değişkenleri “display x.l, y.l, z.l” komutuyla görüntüleyebiliriz.

Burada;

.l = level ya da primal değerdir.

Diğer gösterimler ise şunlardır:

.lo = alt sınır (lower bound)

.up = üst sınır (upper bound)

.m = marjinal ya da dual değer

```
integer variables
```

```
y
```

```
x(i);
```

```
equations
```

```
amac
```

```
KesimSaati
```

```
TenekePlakaSayisi
```

```
GovdeSayisi
```

```
UcSayisi;
```

```
amac..    z =e= p*y - c*sum(i,a(i)*x(i)) - hg*sum(i,g(i)*x(i)) - hu*sum(i,u(i)*x(i));
```

```
KesimSaati..    sum(i, k(i)*x(i)) =l= cm;
```

```
TenekePlakaSayisi..    sum(i, x(i)) =l= oq;
```

```
GovdeSayisi..    sum(i, g(i)*x(i))=g= y;
```

```
UcSayisi..    sum (i, u(i)*x(i)) =g= 2*y;
```

```
model ornek2/all/;
```

```
solve ornek2 using mip maximizing z;
```

```
display x.l, y.l, z.l;
```


Optimal değerler;

$x_1 = 0, x_2 = 24, x_3 = 0, x_4 = 3, y = 60, z = 409.500$

```
**** REPORT SUMMARY :      0      NONOPT
                          0 INFEASIBLE
                          0 UNBOUNDED
```

```
QGAMS Rev 141  Intel /MS Window                      11/30/10 04:39:53 Page 6
General Algebraic Modeling System
Execution
```

```
----- 47 VARIABLE x.L
```

```
2 24.000,    4  3.000
```

```
----- 47 VARIABLE y.L          =          60.000
          VARIABLE z.L          =          409.500
```

Sunucu Ulaştırma Problemi

Bu problemde, İstanbul ve Ankara'da bulunan iki üretim merkezinden, Sakarya, Bolu ve Eskişehir'de bulunan üç pazarın gereksinimlerine göre bilgisayar sunucusu gönderilmesi gereklidir.

	Pazarlar			Kapasite
	Sakarya	Bolu	Eskişehir	
İstanbul	150 km	250 km	400 km	350 adet
Ankara	300 km	200 km	150 km	600 adet
Talep miktarı	325 adet	300 adet	275 adet	

Her bir sunucunun ulaştırma fiyatı gönderim yapılan uzaklığa bağlı olarak km başına 90 TL olarak verilmiştir. Problemin amacı her bir üretim merkezinin kapasitelerini aşmadan her pazarın gereksinimini en düşük maliyetle karşılayacak çözümün bulunmasıdır.

Problemde her fabrika ve pazar endeks olarak tanımlanabilir.

Sets

i fabrikalar /ist, ank/
 j pazarlar /sak, bol, esk/;

Artık tüm değişkenler bu endekslere bağlı olarak tanımlanacaktır.

Parameters

$K(i)$ her fabrikanın kapasitesi /ist 350, ank 600/
 $T(j)$ her pazarın talebi /sak 325, bol 300, esk 275/;

Km başına taşıma maliyeti 90 TL olarak verilmiştir. Bu durumda sabit birim km taşıma maliyeti:

Scalar

m km başına taşıma maliyeti /90/;

u_{ij} : i fabrikasından j pazarına olan uzaklık olmak üzere çift indisli “ u_{ij} ” değişkenini tanımlamak için tablo kullanmalıyız.

Table $u(i,j)$

	sak	bol	esk
ist	150	250	400
ank	300	200	150 ;

Bu durumda birim taşıma maliyeti aşağıdaki şekilde tanımlanır.

Parameter $c(i,j)$ her bir durum için taşıma maliyeti ;
 $c(i,j) = m \cdot u(i,j)$;

olur.

Değişken olarak her fabrikadan her pazara gönderilen sunucu miktarını alabiliriz.

Variables

$x(i, j)$

z ;

Positive variables x ;

Equations

maliyet

kapasite(i)

talep(j) ;

```
maliyet..      z=e=sum((i,j), x(i,j)*c(i,j));  
kapasite(i).. sum(j, x(i,j))=l= K(i);  
talep(j)..    sum(i, x(i,j))=g= T(j);
```

Son olarak modelin çözümü için aşağıda verilen komutları yazmalıyız.

```
Model ornek3 /all/ ;  
Solve ornek3 using lp minimizing z ;
```

İlgilendiğimiz değişkeni aşağıda verilen komutu kullanarak görüntüleyebiliriz.

```
Display x.l, z.l;
```

```

sets i fabrikalar /ist, ank/
j pazarlar /sak, bol, esk/;

parameters
K(i) her fabrikanin kapasitesi /ist 350, ank 600/
T(j) her pazarin talebi /sak 325, bol 300, esk 275/

scalar m km basina tasima maliyeti /90/;

table u(i,j)
      sak bol esk
ist 150 250 400
ank 300 200 150;

parameter c(i,j) her bir durum icin tasima maliyeti;
c(i,j)=m*u(i,j);

variables x(i,j)
z;

positive variable
x;

equations
maliyet
kapasite(i)
talep(j);

maliyet..      z=e=sum((i,j), x(i,j)*c(i,j));
kapasite(i)..  sum(j, x(i,j))=l= K(i);
talep(j)..     sum(i, x(i,j))=g= T(j);

model ornek3/all/;

solve ornek3 using lp minimizing z;

display x.l , z.l;

```

```

LP Presolve eliminated 1 rows and 1 columns.
Reduced LP has 5 rows, 6 columns, and 12 nonzeros.
Presolve time = 0.00 sec.
Initializing dual steep norms . . . .

Iteration      Dual Objective      In Variable      Out Variable
    1          26100000.000000      x(ank.bol)      talep(bol) slack
    2          25987500.000000      x(ist.bol)      talep(sak) slack

Optimal solution found.
Objective : 25987500.000000

--- Restarting execution
--- Ornek3.gms(35) 0 Mb
--- Reading solution for model ornek3
--- Executing after solve
--- Ornek3.gms(37) 3 Mb
*** Status: Normal completion

```


General Algebraic Modeling System Execution

---- 37 VARIABLE x.L

	sak	bol	esk
ist		25.000	325.000
ank	325.000	275.000	

---- 37 VARIABLE z.L = 2.598750E+7

EXECUTION TIME = 0.000 SECONDS 2.9 Mb WIN216-141 Jan 24, 2005

USER: Centre Versailles - Grigon S021106:1507AB-WIN
I.N.R.A. DC2726
License for teaching and research at degree granting institutions

**** FILE SUMMARY

Input C:\Documents and Settings\User\Belgelerim\ITU Course Notes\OR\GAMS\Ornek3.gms

Output C:\WINDOWS\gamsdir\Ornek3.lst

“Bin Packing” Problemi

Bin packing problemi (BPP), bir çok yöneylem araştırması probleminin modellenmesi ve çözümüne temel teşkil eden bir problemidir.

Yapı olarak Knapsack problemine benzemekle birlikte, ondan farklı olarak birden fazla sırt çantası olduğu düşünülebilir. Dolayısıyla amaç ve kısıtlar farklıdır.

Amaç fonksiyonu ve kısıtlara bağlı olarak çok farklı türleri olmakla birlikte, aşağıda BPP’nin en yalın halde amaç fonksiyonu ve kısıtları verilmiştir:

Amaç fonksiyonu:

- *Tüm ürünleri/eşyaları yerleştirebilmek kaydıyla, çanta/koli/kaynak sayısını minimize etmektir.*

Karar Değişkeni:

- *Hangi ürünün hangi koliye yerleştirileceğine karar verilmesi gerekir.*

Problemin kısıtları aşağıdaki şekilde özetlenebilir:

- Tüm ürünler tam olarak bir koliye atanmak zorundadır (atanmamış ürün kalamaz ve bir ürün birden fazla koliye atanamaz).
- Bir koliye atanan ürünlerin ağırlıkları toplamı, kolinin taşıyabileceği maksimum ağırlıktan fazla olamaz.
- Bir koli, ancak içine eşya atanmışsa açılabilir/kullanılabilir/çözüme dahil edilebilir.
- İşaret kısıtları (probleme bağlıdır).

Aşağıdaki kısıt isteğe bağlıdır:

- Koliler, 1 numaralı koliden başlamak üzere açılırlar.

Örnek:

Bir firma, müşterilerine havayolu ile göndereceği 20 adet ürünü, standart boyutlardaki kolilere paketlemek istemektedir. Gönderilecek ürünlere ilişkin ağırlık bilgisi aşağıdaki gibidir:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
W_i	5	7	3	5	6	2	4	4	7	4	8	2	7	1	2	5	4	6	4	3

i : ürün, W_i : i ürününün ağırlığı (kg)

Her bir kolinin taşıyabileceği maksimum ağırlık 10kg'dır.

Hazırlanacak koli sayısını minimize etmek amacıyla, hangi ürünün hangi koliye yerleştirilmesi gerektiğine karar veriniz.

Matematiksel Model (Notasyon):

i : ürün indeksi

j : koli indeksi

w_i : i ürününün ağırlığı (kg)

x_{ij} : 1, i ürünü j kolisine atanmışsa; 0 değilse

z_j : 1, j kolisi kullanılmışsa (en az bir ürün atanmışsa); 0 değilse

KP : Her bir kolinin taşıyabileceği maksimum ağırlık (kg)

Matematiksel Model (Amaç Fonksiyonu ve Kısıtlar):

$$\text{Min } OBJ = \sum_j z_j$$

$$\sum_j x_{ij} = 1 \quad \forall i$$

-> Her ürün tam olarak bir koliye atanmış olmalıdır.

$$\sum_i x_{ij} w_i \leq KP \quad \forall j$$

-> Bir koliye atanacak ürünlerin ağırlığı toplamı, kolinin taşıyabileceği maksimum ağırlığı aşamaz.

$$x_{ij} \leq z_j \quad \forall i, j$$

-> Bir koli, ancak bu koliye en az bir ürün atanmışsa açılabilir.

$$z_{(j+1)} \leq z_j \quad \forall j$$

-> Koliler, ilk koliden başlamak üzere açılırlar.

$$x_{ij}, z_j \in \{0,1\}$$

-> İşaret kısıtları

```

set
i urun /1*20/
j koli /1*20/
;

parameter
w(i) ürün ağırlıkları (kg)
/1 5
 2 7
 3 3
 4 5
 5 6
 6 2
 7 4
 8 4
 9 7
10 4
11 8
12 2
13 7
14 1
15 2
16 5
17 4
18 6
19 4
20 3
/

scalar
KP her bir kolinin taşıyabileceği maksimum ağırlık (kg) /10/
;

```

```

variable
x(i,j) i urunu j kolisine atanmışsa
z(j) j kolisi kullanılmışsa
OBJ amaç fonk
;

binary variable
x(i,j)
z(j)
;

equation
amac
atamakisit
kapasitekisit
koliacmakisit
kolisirakisit
;

amac.. OBJ=e=sum(j, z(j));
atamakisit(i).. sum(j, x(i,j)) =e= 1;
kapasitekisit(j).. sum(i, x(i,j)*w(i)) =l= KP;
koliacmakisit(i,j).. x(i,j) =l= z(j);
kolisirakisit(j).. z(j+1) =l= z(j);

model MBP /all/;
solve MBP using mip minimizing OBJ;

display x.l, kapasitekisit.l;

```

---- 66 VARIABLE x.L i urunu j kolisine atanmışsa

	1	2	3	4	5	6
5			1.000			
6				1.000		
7				1.000		
9						1.000
10				1.000		
11	1.000					
12		1.000				
13		1.000				
14		1.000				
15	1.000					
17					1.000	
18					1.000	
19			1.000			
20						1.000
+	7	8	9			
1	1.000					
2		1.000				
3		1.000				
4			1.000			
8			1.000			
16	1.000					

Toplamda 9 adet koli kullanılmıştır.

Koli	Atanan Ürünler
1	11, 15
2	12, 13, 14
3	5, 19
4	6, 7, 10
5	17, 18
6	9, 20
7	1, 16
8	2, 3
9	4, 8

---- 66 EQUATION kapasitekisit.L

1 10.000, 2 10.000, 3 10.000, 4 10.000, 5 10.000, 6 10.000
7 10.000, 8 10.000, 9 9.000

Koli	Atanan Ürünler	Ağırlık (kg)
1	11, 15	10
2	12, 13, 14	10
3	5, 19	10
4	6, 7, 10	10
5	17, 18	10
6	9, 20	10
7	1, 16	10
8	2, 3	10
9	4, 8	9

Çalışma Soruları

Matematiksel modellemesi verilen problemleri GAMS kullanarak çözünüz.

Soru 1

Bir işletme, aşağıdaki özellikleri taşıyan ürün üretmektedir:

$$\text{Yanma Noktası} \geq 2800$$

$$\text{Özgül Ağırlığı} \geq 1.00$$

$$\alpha \text{ muhtevası} \leq \% 8 \text{ hacim olarak}$$

$$\beta \text{ muhtevası} \leq \% 3 \text{ hacim olarak}$$

Ürün, litresi sırası ile 16 TL, 20 TL ve 25 TL olan A, B ve C yarı mamullerinden elde edilebilmektedir. Bu üç yarı mamulün özellikleri aşağıdaki gibidir.

	A	B	C
Yanma noktası	2800	2700	2900
Özgül ağırlığı	0.95	1.20	1.10
α muhtevası (%)	7	8	11
β muhtevası (%)	4	5	3

Üretici, yukarıdaki koşullara uygun olarak ürününü en düşük maliyetle üretmek istemektedir. Buna göre problemin doğrusal programlama modeli nasıl kurulur ve çözülür?

Matematiksel Model:

Karar Değişkenleri

x_1 = Üründeki A oranı

x_2 = Üründeki B oranı

x_3 = Üründeki C oranı

Amaç fonksiyonu

$$\text{Min } Z = 16x_1 + 20x_2 + 25x_3$$

Problem Kısıtları

$$x_1 + x_2 + x_3 = 1$$

$$2800x_1 + 2700x_2 + 2900x_3 \geq 2800 \quad (\text{Yanma noktası kısıtı})$$

$$0.95x_1 + 1.20x_2 + 1.10x_3 \geq 1.00 \quad (\text{Özgül ağırlık kısıtı})$$

$$7x_1 + 8x_2 + 11x_3 \leq 8 \quad (\alpha \text{ muhtevası kısıtı})$$

$$4x_1 + 5x_2 + 3x_3 \leq 3 \quad (\beta \text{ muhtevası kısıtı})$$

Soru 2

W7ABC.gms

ABC şirketi yeni bir yer silme sıvısı üretecektir. Bu sıvının içinde A temizleyici maddesinden en az %20 en fazla %40, B temizleyici maddesinden en az %25 en fazla %50 ve C temizleyici maddesinden en az %35 en fazla %50 olacaktır. Şirket, A, B ve C maddelerini üretmek için ise 3 ayrı hammadde kullanmaktadır. Bu maddelerin içerikleri aşağıdaki tabloda verilmiştir.

Hammadde	A maddesi	B maddesi	C maddesi
I	%60	%30	%10
II	%25	%25	%50
III	%20	%50	%30

Hammadde I, II ve III' ün kg maliyetleri sırasıyla 1.5 TL, 1.55 TL, 1.6 TL'dir. Şirket en az maliyetle talebi karşılayacak yer silme sıvısının üretimi için doğrusal programlama modeli oluşturup çözmek istemektedir.

Karar değişkenleri

Karışım probleminde, her biri ürün içinde yer alacak hammadde miktarına karşılık gelen 3 karar değişkeni tanımlanmalıdır. Bunlar;

P_i = ürünün içinde yer alacak i hammaddesi oranı

Amaç fonksiyonu

Problemin amacı kullanılan hammadde maliyetini minimize etmektir. Dolayısıyla, karar değişkenleri ile o karar değişkenine karşılık gelen maliyet değerleri çarpılıp sonra da tüm değerler toplanarak amaç fonksiyonu elde edilir.

$$\text{Min } Z = 1.5 P_1 + 1.55P_2 + 1.6P_3$$

Problemin kısıtları

Problemdeki ilk üç kısıt, ürün içinde A, B ve C maddelerinden bulunması gereken en az miktarı sağlayan kısıtlardır.

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \geq 0.20$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \geq 0.25$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \geq 0.35$$

Problemdeki ikinci grup üç kısıt da, ürünün içinde A, B ve C maddelerinden bulunması gereken en fazla miktarı sağlayan kısıtlardır.

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \leq 0.4$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \leq 0.5$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \leq 0.5$$

Üç hammaddeden kullanılan oranların toplamının %100 olmasını sağlayan kısıt da aşağıdaki şekilde yazılmalıdır.

$$P_1 + P_2 + P_3 = 1$$

Son olarak karar değişkenlerinin negatif olmama kısıtları da aşağıda gösterildiği gibi modele eklenmelidir.

$$P_i \geq 0$$

Amaç fonksiyonu ve kısıtlar bir araya getirilerek karışım modeli aşağıdaki gibi oluşturulur.

$$\text{Min } Z = 1.5 P_1 + 1.55P_2 + 1.6P_3$$

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \geq 0.20$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \geq 0.25$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \geq 0.35$$

$$0.60 P_1 + 0.25P_2 + 0.20P_3 \leq 0.40$$

$$0.30 P_1 + 0.25P_2 + 0.50P_3 \leq 0.50$$

$$0.10 P_1 + 0.50P_2 + 0.30P_3 \leq 0.50$$

$$P_1 + P_2 + P_3 = 1$$

$$P_i \geq 0$$

Soru 3

Bir rafineri 2 tür kurşunsuz benzin üretimi yapmaktadır. 1. tipin varil fiyatı 48\$, 2. tipin varil fiyatı ise 53\$'dır. Her iki tip benzin de Tablo 1'de verilen özellikleri karşılamak zorundadır. Karışımında kullanılan bileşenler ve özellikleri de Tablo 2'de verilmiştir. Haftalık karı maksimize etmek için iki tip benzine hangi bileşenlerden hangi miktarlarda karıştırılmalıdır? Geliştirilen matematiksel modele göre Excel Solver kullanarak çözümü bulunuz.

Tablo 1

Benzin tipi	Minimum oktan oranı	Maksimum talep (varil/hafta)	Minimum dağıtım (varil/hafta)
1. Tip Benzin	87	80.000	60.000
2. Tip Benzin	93	40.000	15.000

Tablo 2

Benzin Bileşenleri	Oktan oranı	Arz miktarı (varil)	Maliyet (\$/varil)
1	86	70.000	33
2	96	60.000	37

Karar değişkenleri

Bu problemdeki kontrol değişkenleri, benzinlere karıştırılması gereken iki bileşenin miktarlarıdır.

x_{ij} = j tip benzine karıştırılan i bileşeninin varil cinsinden haftalık miktarı

$$i = 1, 2 ; \quad j = 1, 2$$

Amaç Fonksiyonu

Benzin bileşenlerinin doğrusal olarak karıştığı kabul edilirse, 1. tip benzin miktarı $x_{11} + x_{21}$, ikinci tip benzin miktarı ise $x_{12} + x_{22}$ olarak verilir. Benzer şekilde kullanılan 1. ve 2. bileşenlerin toplam miktarları da sırasıyla $x_{11} + x_{12}$ ve $x_{21} + x_{22}$ olarak verilebilir. Amaç fonksiyonu şu şekilde formüle edilebilir:

$$Kar = \text{satış} - \text{maliyet}$$

$$Max Z = 48(x_{11} + x_{21}) + 53(x_{12} + x_{22}) - 33(x_{11} + x_{12}) - 37(x_{21} + x_{22}) \quad \text{haftalık kar}$$

$$Max Z = 15x_{11} + 20x_{12} + 11x_{21} + 16x_{22}$$

Problemin Kısıtları

Minimum oktan oranı ihtiyacını karşılamak için, iki bileşenin değişik miktarlarda karıştırılması durumunda ortaya çıkacak oktan seviyesini belirlemeliyiz. Bileşenlerin yine doğrusal olarak karıştığını kabul ediyoruz. Bileşenlerin karışımındaki oktan seviyesi ağırlıklı ortalama alınarak hesaplanabilir. Karışımındaki toplam oktanı, varil miktarına bölerek buluruz.

Böylece, 1. tip benzin için minimum oktan oranı kısıtı

$$\frac{86x_{11} + 96x_{21}}{x_{11} + x_{21}} \geq 87$$

Bu kısıt lineer değildir fakat eşitsizliğin her iki tarafını

$x_{11} + x_{21}$ ile çarparsak;

$$-x_{11} + 9x_{21} \geq 0$$

Aynı şekilde, 2. tip benzin için minimum oktan oranı kısıtı

$$\frac{86x_{12} + 96x_{22}}{x_{12} + x_{22}} \geq 93$$

Bu kısıt lineer değildir fakat eşitsizliğin her iki tarafını

$x_{12} + x_{22}$ ile çarparsak;

$$-7x_{12} + 3x_{22} \geq 0$$

Benzin tipleri için minimum ve maksimum dağıtım gereksinimi kısıtları ise şu şekilde formüle edilebilir:

$$x_{11} + x_{21} \geq 60.000 \quad (1. \text{tip benzin için minimum dağıtım miktarı})$$

$$x_{12} + x_{22} \geq 15.000 \quad (2. \text{tip benzin için minimum dağıtım miktarı})$$

$$x_{11} + x_{21} \leq 80.000 \quad (1. \text{tip benzin için maksimum talep miktarı})$$

$$x_{12} + x_{22} \leq 40.000 \quad (2. \text{tip benzin için maksimum talep miktarı})$$

Benzer şekilde, bileşenlerin arz kısıtları:

$$x_{11} + x_{12} \leq 70.000 \quad (1. \text{bileşenin arz miktarı})$$

$$x_{21} + x_{22} \leq 60.000 \quad (2. \text{bileşenin arz miktarı})$$

Negatif olmama kısıtları:

$$x_{11}, x_{12}, x_{21}, x_{22} \geq 0$$

Böylece lineer programlama modelinin tamamı aşağıdaki şekilde yazılabilir.

$$\text{Max } Z = 15x_{11} + 20x_{12} + 11x_{21} + 16x_{22}$$

$$-x_{11} + 9x_{21} \geq 0$$

$$-7x_{12} + 3x_{22} \geq 0$$

$$x_{11} + x_{21} \geq 60.000$$

$$x_{12} + x_{22} \geq 15.000$$

$$x_{11} + x_{21} \leq 80.000$$

$$x_{12} + x_{22} \leq 40.000$$

$$x_{11} + x_{12} \leq 70.000$$

$$x_{21} + x_{22} \leq 60.000$$

$$x_{11}, x_{12}, x_{21}, x_{22} \geq 0$$

Doğrusallık varsayımları altında, modelimiz karışım gereksinimlerini tam olarak sağlamaktadır. Fakat satışlardan elde edilen gelir ve maliyetler, amaç fonksiyonunda görünmemektedir.

1. ve 2. tip benzin miktarlarına sırasıyla g_1 ve g_2 dersek;

$$g_1 = x_{11} + x_{21} \text{ veya}$$

$$x_{11} + x_{21} - g_1 = 0 \text{ (1. tip benzinin varil cinsinden toplam miktarı)}$$

Aynı şekilde;

$$g_2 = x_{12} + x_{22} \text{ veya}$$

$$x_{12} + x_{22} - g_2 = 0 \text{ (2. tip benzinin varil cinsinden toplam miktarı)}$$

1. ve 2. bileşenlere ise sırasıyla h_1 ve h_2 dersek;

$$x_{11} + x_{12} - h_1 = 0 \text{ (1. tip bileşenin varil cinsinden toplam miktarı)}$$

$$x_{21} + x_{22} - h_2 = 0 \text{ (2. tip bileşenin varil cinsinden toplam miktarı)}$$

Bu yeni ilişkilerden yararlanarak önceki modeli daha basit bir şekilde ifade edebiliriz.

Amaç Fonksiyonu:

$$\text{Maksimum kar } Z = 48g_1 + 53g_2 - 33h_1 - 37h_2$$

Kısıtlar:

$$x_{11} + x_{21} - g_1 = 0 \qquad g_1 \geq 60.000$$

$$x_{12} + x_{22} - g_2 = 0 \qquad g_2 \geq 15.000$$

$$x_{11} + x_{12} - h_1 = 0 \qquad g_1 \leq 80.000$$

$$x_{21} + x_{22} - h_2 = 0 \qquad g_2 \leq 40.000$$

$$-x_{11} + 9x_{21} \geq 0 \qquad h_1 \leq 70.000$$

$$-7x_{12} + 3x_{22} \geq 0 \qquad h_2 \leq 60.000$$

$$x_{11}, x_{12}, x_{21}, x_{22}, g_1, g_2, h_1, h_2 \geq 0$$

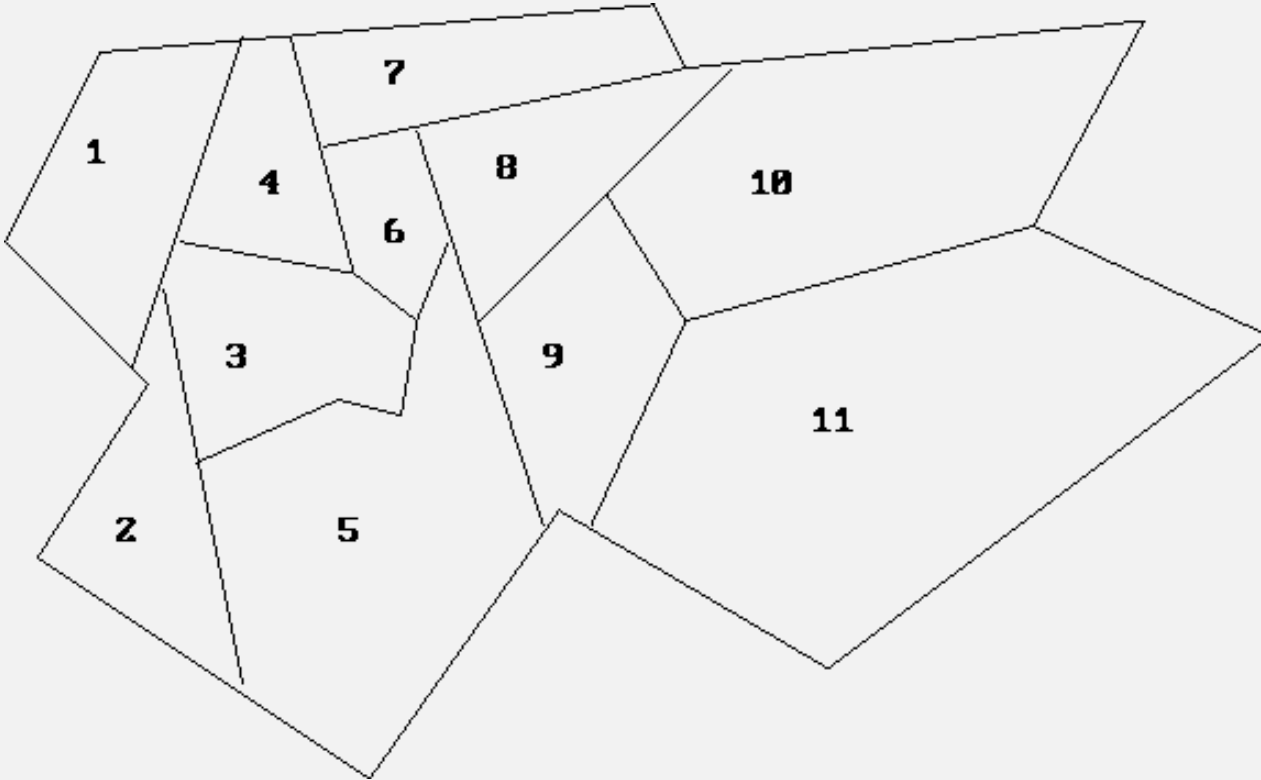
Küme Kapsama Problemi

– Örnek ve Çözümü

7

Küme Kapsama (Set Covering)

Bu problemi anlatmak için aşağıdaki yer belirleme problemini düşünelim.



Şekilde verilen haritada numaralandırılan mahallelere hizmet edecek itfaiye istasyonlarının yerleri gözden geçirilmek/yeniden belirlenmek isteniyor. Bir mahalleye kurulacak olan itfaiye istasyonu sadece komşuluğunda bulunan mahallelere hizmet verebiliyor. Örneğin 11 numaralı mahalleye kurulacak olan bir istasyon, sadece 9, 10 ve 11 numaralı mahallelere hizmet verebilir.

Küme Kapsama (Set Covering)

x_j , j bölgesine itfaiye istasyonu kurulmuşsa 1, aksi halde sıfır değerini alan bir karar değişkeni olarak düşünülürse problem aşağıdaki gibi modellenenebilir.

$$\begin{aligned} & \text{Minimize } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} \\ & \text{subject to } x_1 + x_2 + x_3 + x_4 \geq 1 \\ & \quad x_1 + x_2 + x_3 + x_5 \geq 1 \\ & \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1 \\ & \quad x_1 + x_3 + x_4 + x_6 + x_7 \geq 1 \\ & \quad x_2 + x_3 + x_5 + x_6 + x_8 + x_9 \geq 1 \\ & \quad x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1 \\ & \quad x_4 + x_6 + x_7 + x_8 \geq 1 \\ & \quad x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \geq 1 \\ & \quad x_5 + x_8 + x_9 + x_{10} + x_{11} \geq 1 \\ & \quad x_8 + x_9 + x_{10} + x_{11} \geq 1 \\ & \quad x_9 + x_{10} + x_{11} \geq 1 \\ & \quad x_j \in \{0,1\} \quad j = 1, \dots, 11 \end{aligned}$$

Küme Kapsama (Set Covering)

Burada amaç fonksiyonu, kurulacak olan itfaiye istasyonu sayısını minimize etmeyi amaçlarken, kısıtlar ise her bir mahallenin komşuluğunda mutlaka en az bir tane itfaiye istasyonunun bulunmasını garanti eder.

Aşağıda bu konuyla ilgili bir örnek verilmiştir.

Küme Kapsama (Set Covering)

W10Kume.gms

There are six cities in region R. The region must determine where to build fire stations. The region wants to build the minimum number of fire stations and ensure that at least one fire station is near 15 minutes of each city. The times (in minutes) required to drive between cities are:

From	To					
	1	2	3	4	5	6
1	0	10	20	30	30	20
2	10	0	25	35	20	10
3	20	25	0	15	30	20
4	30	35	15	0	15	25
5	20	20	30	15	0	14
6	20	10	20	25	14	0

$$x_i = \begin{cases} 1 & \text{if a fire station is built in city } i \\ 0 & \text{otherwise} \end{cases}$$

Küme Kapsama (Set Covering)

Objective function: Total number of fire stations to be built

$$x_0 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

Constraints: A fire station within 15 mins of each city.

The locations that can reach each city in 15 minutes:

$$\begin{array}{ll} \text{IP: min} & x_0 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ \text{subject to} & \begin{array}{ll} x_1 + x_2 & \geq 1 \text{ (City 1)} \\ x_1 + x_2 + x_6 & \geq 1 \text{ (City 2)} \\ x_3 + x_4 & \geq 1 \text{ (City 3)} \\ x_3 + x_4 + x_5 & \geq 1 \text{ (City 4)} \\ x_4 + x_5 + x_6 & \geq 1 \text{ (City 5)} \\ x_2 + x_5 + x_6 & \geq 1 \text{ (City 6)} \end{array} \\ & x_i = 1 \text{ or } 0; i = 1, \dots, 6. \end{array}$$

Optimal solution: $x_0 = 2; x_2 = x_4 = 1, x_1 = x_3 = x_5 = x_6 = 0$.

Küme Kapsama (Set Covering)

Bu problem açık formda çözülmek istenirse yukarıdaki gibi bir yaklaşım izlenebilir. Fakat kapalı formda (set kullanarak) çözüm yapmak için aşağıdaki gibi kurulan bir matematiksel ifadenin kullanılması gerekmektedir.

$$\begin{aligned} &\text{Minimise} && \sum_{j=1}^n x_j \\ &\text{subject to} && \sum_{j=1}^n a_{ij} x_j \geq 1 \quad i = 1, \dots, m \\ &&& x_j = 0 \text{ or } 1 \end{aligned}$$

Burada kullanılan a_{ij} ifadesi, eğer i noktası j noktasına kurulacak olan bir istasyondan hizmet alabiliyorsa 1, aksi halde 0 değerini alır.

a_{ij} değerleri, probleme girdi olarak sağlanabilir (GAMS'te table kullanarak).

```

set
j şehirler /1*6/
;

Alias (i,j);

table a(i,j) hizmet_alabilirlik
  1  2  3  4  5  6
1 1  1  0  0  0  0
2 1  1  0  0  0  1
3 0  0  1  1  0  0
4 0  0  1  1  1  0
5 0  0  0  1  1  1
6 0  1  0  0  1  1
;

```

```

variable
x(j)
z;

binary variable
x(j)
;

equations
amac
kisit1
;

amac.. z =e= sum(j, x(j));
kisit1(i).. sum(j, a(i,j)*x(j)) =g= 1;

model setcover /all/;
solve setcover using mip minimizing z;

```


S O L V E S U M M A R Y

MODEL setcover OBJECTIVE z
 TYPE MIP DIRECTION MINIMIZE
 SOLVER CPLEX FROM LINE 46

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 2.0000

Nihai olarak 2 ve 4 numaralı şehirlere kurulum yapılması gerektiği görülmektedir.

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1	.	.	1.000	1.000
2	.	1.000	1.000	1.000
3	.	.	1.000	1.000
4	.	1.000	1.000	1.000
5	.	.	1.000	1.000
6	.	.	1.000	1.000

	LOWER	LEVEL	UPPER	MARGINAL
--	-------	-------	-------	----------

---- VAR z

	-INF	2.000	+INF	.
--	------	-------	------	---

**** REPORT SUMMARY :

0 NONOPT
 0 INFEASIBLE
 0 UNBOUNDED

Dosyaya Yazdırma ve Dosyadan Okuma

Program Akış Kontrolü

Subsets, Ordered Sets, Dynamic Sets

sameas ve diag, \$ Operatörü

8

Kaynak: Dhazn Gillig and Bruce A. McCarl, Introduction to GAMS: Summation Notation with GAMS, Department of Agricultural Economics, Texas A&M University (*from Appendix A in McCarl and Spreen's text book on* <http://ageco.tamu.edu/faculty/mccarl/mccspr/newa1.pdf>)

Subset (Alt Küme)

Söz dizimi:

$\text{set1} \subseteq \text{set2}$: `set set1 (set2);`

► Example:

```
set ice / chocolate, strawberry, cherry, vanilla /;  
set sorbet(ice) / strawberry, cherry /;
```

► Domain checking: `set sorbet(ice) / strawberry, banana /;` $\Rightarrow$ **error**

Cardinality (Kümedeki Eleman Sayısı)

Card(x) komutu x kümesindeki eleman sayısını verir.

► Example:

```
set c      'countries' / jamaica, haiti, guyana, brazil /;  
scalar nc 'number of countries;  
nc = card(c);
```

ord() komutu

Bir kümedeki bir elemanın o küme içindeki sıra numarasına ihtiyaç olursa “ord(x)” fonksiyonu kullanılabilir.

Örneğin;

$$y = \sum_i i * x_i$$

ifadesi GAMS'te aşağıdaki gibi kodlanabilir:

```
y =e= sum(i, ord(i) *X(i) )
```

► Example:

```
Set t / 1*24 /;  
Parameter hour(t);  
hour(t) = ord(t);
```

Küme Elemanlarının Komşulukları (Önceki ve Sonraki Elemanlara Erişim)

Sıralı kümelerde (*ordered sets*), küme elemanının komşuluğundaki öğelere erişmek mümkündür.

Söz dizimi: **setelement $\pm n$**

Eğer erişilmeye çalışılan öğe için kümede bir karşılık yoksa veya küme boyutlarını aşıyorsa, yani “**setelement + n > card(set)**” ise dönecek değer sıfırdır.

Örneğin;

```
Set t /1*24/;
```

```
Variables Stok(t), UretimMik(t), Satis(t);
```

```
Equation Stok;
```

```
StokHesaplama(t) .. Stok(t) =e= Stok(t-1)+UretimMik(t)-Satis(t);
```

Koşullu İfadeler: Boolean (İkili) Operatörler

Sayısal ve mantıksal operatörler aşağıda verildiği gibi kullanılır.

İkili operatörlerin işlenmesi sonucu true dönerse 1'e, false dönerse 0'a karşılık gelmektedir.

Sayısal operatörler

GAMS	Anlamı
lt	<
le	<=
eq	=
ne	<>
ge	>=
gt	>

Mantıksal operatörler

GAMS	Anlamı
not	Değil
and	Ve
or	Veya
xor	Farklı veya

<i>a</i>	<i>b</i>	<i>a and b</i>	<i>a or b</i>	<i>a xor b</i>	<i>not a</i>
0	0	0	0	0	1
0	non-zero	0	1	1	1
non-zero	0	0	1	1	0
non-zero	non-zero	1	1	0	0

Koşullu İfadeler: \$ Operatörü

- \$ operatörü gerekli koşulların uygulanmasını mümkün kılar.
- \$ (**koşul**) komutu “eğer koşul doğru ise” olarak düşünülebilir.

- Örnek:

Eğer $b > 1.5$ ise $a = 2$ olsun -> $a \$ (b > 1.5) = 2 ;$

Eğer $b > 1.5$ ise $a = 2$ olsun, aksi halde $a = 0$ olsun -> $a = 2 \$ (b > 1.5) ;$

- \$ eşitliğin solunda ise, koşul sağlanmazsa atama yok
- \$ eşitliğin sağında ise, her zaman atama yapılır. Fakat koşul sağlanmazsa \$ ile belirtilen ifade 0 olur.
- \$ ifadesi tanımlamalar için kullanılamaz.

Koşullu İfadeler: \$ Operatörü

Kısıtları bazı koşullara göre süzmek için yine \$ işareti kullanılabilir.

Örnek:

Aynı iki indise sahip x_{ij} ve y_{ij} değişkenleri olsun. Sadece i 'nin j 'den küçük olduğu durumlar için, y_{ij} 'nin x_{ij} 'den büyük olması gerektiğini, diğer durumlar için bir kısıt olmadığını varsayalım. Bu kısıtı matematiksel olarak aşağıdaki gibi yazabiliriz:

$$y_{ij} \geq x_{ij} \quad \forall i < j$$

Bu matematiksel ifadenin GAMS'teki karşılığı ise aşağıdaki gibi olur:

```
Kisit1 (i , j) $ (ord (i) < ord (j) ) .. y (i , j) =g= x (i , j) ;
```

\$ Operatörü Uygulamaları

Filtering in indexed operations:

```
parameter sorbetbalance;  
set ice; set sorbet(ice);  
sorbetbalance = sum(ice$sorbet(ice), price(ice)*purchase.l(ice));  
sorbetbalance = sum(sorbet(ice), price(ice) * purchase.l(ice));
```

Conditioned indexed operations:

```
rho = sum(i$(sig(i) ne 0), 1/sig(i) - 1);
```

Conditioned equations:

```
Equation satdemand(ice);  
satdemand(ice)$sorbet(ice).. purchase(ice) =g= demand(ice);  
  
balance(t)$(ord(t)>1)..  
level(t) =e= level(t-1) + inflow(t) - outflow(t);
```

Existence of variables in constraints:

```
variables x, y; parameter A; equation e;  
e.. x + y$(A>2) =e= A;
```

\$ Operatörü İçinde Değişken KULLANILAMAZ!

- ▶ The following **DOES NOT WORK**:

```
binary variable x;  variable y;  equation e1, e2;  
e1$(x = 1).. y      =l= 100;  
e2          .. y$(x = 1) =l= 100;
```

- ▶ The value of **x** is decided by the solver, not by the GAMS.
- ▶ However, GAMS has to evaluate \$-operators when assembling an instance in the Solve statement.
- ▶ Instead, you have to **reformulate**:

```
e.. y =l= 100 * x + y.up * (1-x);
```

That is: $x=1 \Rightarrow y \leq 100$; $x=0 \Rightarrow y \leq y.up$.

sameas ve diag

sameas (*i1*, *i2*) komutu, *i1* ve *i2*'nin birbirine eşit olma durumunu kontrol eder. Eşit olması durumunda **true**, olmaması durumunda ise **false** döndürür.

Benzer şekilde, *i1*'in verilen "**değer**"e eşit olma durumunu kontrol etmek için de **sameas** (*i1*, "**değer**") komutu kullanılır.

diag komutu da tıpkı **sameas** gibi kullanılır, ancak eşitlik durumunda **1** değerini, eşitsizlik durumunda ise **0** değerini döndürür.

Örnek:

```
sum ( (i, j) $ (sameas (i, j)) , C (i, j)
```

Burada toplama işlemi sadece *i*'nin *j*'ye eşit olduğu **C (i, j)** değerleri için yapılır.

$\text{sum}((i, j) \$ (\text{NOT sameas}(i, j)), C(i, j))$

Komutunda ise toplama işlemi sadece i 'nin j 'den farklı olduğu $C(i, j)$ değerleri için yapılır. Aşağıdaki örnek Chicago'dan tekrar Chicago'ya gönderimi engellemektedir.

SETS

Source

/ Chicago
"San Diego"/

Destination

/ "New York"
Chicago
Topeka / ;

```
Costsum ..  
TotalCost  
=E= SUM((Source, Destination)  
      $ (not sameas(Source, Destination)),  
        Trancost(Source, Destination)  
        *Transport(Source, Destination));  
SupplyBal(Source)..  
SUM(Destination$ (not sameas(Source, Destination)),  
     Transport(Source, Destination) )  
=L= Supply(Source) ;  
DemandBal(Destination) ..  
SUM(Source$ (not sameas(Source, Destination)),  
     Transport(Source, Destination) )  
=G= Need(Destination) ;
```

```
z(k) = sum(i$sameas(i, 'water'), r(i,k));
```

şeklindeki bir komut, sadece i 'nin water değerini aldığı “ $r(i,k)$ ” değerlerinin toplamını her bir k değeri için $z(k)$ ’ya eşitlemektedir.

Yani i 'nin water değerini almadığı diğer “ $r(i,k)$ ” değerleri için bu kısıt aktif değildir.

sameas ve **diag**, iki kümenin birbirine eşit olup olmadığını sorgulamak için de kullanılabilir.

Örnek:

```
sets ice1 / chocolate, strawberry, cherry, vanilla /  
      ice2 / strawberry, cherry, banana /;  
scalar ncommon;  
ncommon = sum((ice1, ice2), diag(ice1,ice2));  
ncommon = sum(sameas(ice1, ice2), 1);
```

Text Dosyasına Yazdırma

- **\$Echo** ve **\$onEcho** - **\$offEcho** komutları text dosyasına yazdırmayı sağlar. Örnek:

```
$Echo "merhaba" > dosya1.txt
```

veya

```
$OnEcho >> dosya1.txt
```

```
Yeni bir mesaj daha
```

```
$OffEcho
```

- **"> dosya1.txt"** komutu **dosya1.txt** isminde yeni bir text dosyası oluşturur. Eğer aynı isimde bir dosya varsa üzerine yazar (override).
- **">> dosya1.txt"** komutu ise varolan dosya1.txt isimli text dosyasına metin ekler.
- Hatırlatma: Bu komut derleme zamanı komutudur. Çözüm sonuçlarını dosyaya yazdırmak için kullanılmaz. Sonraki derslerde bunun için **put** özelliğini kullanmayı öğreneceğiz.

Dosyadan Okuma (**\$Include Komutu**)

- **\$Include** komutu GAMS programına ASCII dosyalardan veri eklemek için kullanılır.
- Sonrasında program, dahil edilen veriyi kullanarak çalışmaya devam eder. Örnek:

```
Table d(i,j) şehirler arasındaki mesafe  
$include dist.txt
```

Burada dist.txt dosyası aşağıdaki tabloyu içermektedir:

	new-york	chicago	topeka
seattle	2.5	1.7	1.8
san-diego	2.5	1.8	1.4 ;

Dosyadan Okuma (\$OnDelim Komutu)

- **\$OnDelim** komutu, virgülle ayrılmış değerler (comma separated values – .csv) formatında, table ve parameter girişi yapmak için kullanılabilir.

- Örnekler:

```
Table d(i,j) uzaklık
$ondelim
$include uzaklik.csv
$offdelim
;
```

Burada uzaklik.csv dosyasının içeriği:

```
, new-york, chicago, topeka
seattle, 2.5, 1.7, 1.8
san-diego, 2.5, 1.8, 1.4
```

```
Parameter d(i,j) uzaklık/
$ondelim
$include uzaklik.txt
$offdelim
/;
```

Burada uzaklik.txt dosyasının içeriği:

```
seattle, new-york, 2.5
san-diego, new-york, 2.5
Seattle, Chicago, 1.7
san-diego, chicago, 1.8
seattle, topeka, 1.8
san-diego, topeka, 1.4
```

Put Komutu

- Program çalışırken çözüm sırasında (execution time) text dosyasına yazmak için kullanılır.
- Dosya ile birlikte bir de fileid (dosyano) kullanılır: `file fileid /dosya1.txt/;`
- Yazmak için bir veri akım yolu (stream) ve dosya adı seçilir: `put fileid;`
- Text, etiketler, numaralar yazdırmak için: `put item{, item};`
- Boş satır bırakmak için: `put /;`
- Veri akımını kapatmak için: `putclose;`

Örnek:

```
file fx /sonuc.txt/;  
put fx 'Fabrikalardan pazarlara sevkedilen miktarlar' /;  
put '-----' /;  
loop ((i,j)$x.l(i,j);  
    put 'Fabrika ' i.te(i):10, 'dan ' j.te(j):10, 'a  
    sevkedilen: ', x.l(i,j) /;  
);  
putclose;
```

Fabrikalardan pazarlara sevkedilen miktarlar

Fabrika seattle'dan new-york'a sevkedilen: 50.00

Fabrika seattle'dan chicago'a sevkedilen: 300.00

Fabrika san-diego'dan new-york'a sevkedilen: 275.00

Fabrika san-diego'dan topeka'a sevkedilen: 275.00

Gams Program Akışının Kontrolü

Program akışı, GAMS çözüm yaparken yönlendirilebilir/kontrol edilebilir.

Program akışını yönlendirmek için kullanılan komutlar:

loop, if-else, while, for

Bu komut blokları içinde denklem tanımlamalarına veya yazımlarına izin verilmemektedir.

Solve komutu kullanılabilir.

loop Komutu*

Loop komutu döngü oluşturmak için kullanılır. Örnek:

► Syntax:

```
loop(controllingset[$(condition)],  
      statement {; statement}  
    );
```

► Example:

```
set t / 1985*1990 /;  
parameter pop(t) / 1985  3456 /  
           growth(t) / 1985  25.3,  1986  27.3,  1987  26.2,  
                      1988  27.1,  1989  26.6,  1990  26.6 /;  
loop(t, pop(t+1) = pop(t) + growth(t));
```

if – elseif-else Komutu

Koşula bağlı olarak akışı yönlendirmek için kullanılır.

If-elseif-else yapısının söz dizimi aşağıdaki gibidir:

```
if (koşul,  
    çalışacak komutlar;  
elseif şart,  
    çalışacak komutlar;  
else  
    çalışacak komutlar;  
);
```

► Example:

```
if( (ml.modelstat eq 4),  
*   model ml was infeasible,  
*   relax bound and solve again  
    x.up(j) = 2*x.up(j);  
    solve ml using lp min z;  
elseif ml.modelstat eq 1,  
    display x.l;  
else  
    abort "Error solving the model";  
);
```


if – elseif-else Komutu

“\$” kullanılarak yapılan koşullandırma ifadeleri, if-elseif-else yapısı ile de ifade edilebilirler.

Örnek:

```
p(i)$(f <= 0) = -1 ;  
p(i)$((f > 0) and (f < 1)) = p(i)**2 ;  
p(i)$(f > 1) = p(i)**3 ;  
q(j)$(f <= 0) = -1 ;  
q(j)$((f > 0) and (f < 1)) = q(j)**2 ;  
q(j)$(f > 1) = q(j)**3 ;
```

```
if (f <= 0,  
    p(i) = -1 ;  
    q(j) = -1 ;  
elseif ((f > 0) and (f < 1)),  
    p(i) = p(i)**2 ;  
    q(j) = q(j)**2 ;  
else  
    p(i) = p(i)**3 ;  
    q(j) = q(j)**3 ;  
) ;
```

while Komutu

while ve **repeat** yapıları döngü oluşturmak için kullanılır.

while komutu söz dizimi:

```
while (koşul,  
      çalışacak komutlar;  
);
```

► Example:

```
scalar count / 1 /;  
scalar globmin / inf /;  
while(count le 1000,  
      x.l(j) = uniform(x.lo(j), x.up(j));  
      solve m1 using nlp min obj;  
      if ( obj.l le globmin, globmin = obj.l; );  
      count = count + 1;  
);
```

while Komutu

repeat komutu söz dizimi:

```
repeat(  
    çalışacak komutlar;  
until koşul);
```

► Example:

```
scalar count / 1 /;  scalar globmin / inf /;  
repeat( x.l(j) = uniform(x.lo(j), x.up(j));  
        solve m1 using nlp min obj;  
        if ( obj.l le globmin, globmin = obj.l; );  
        count = count + 1;  
until count eq 1000 );
```

For Döngüsü

Tıpkı **while** ve **repeat** gibi, döngü oluşturmak için **for** yapısı da kullanılabilir.

Söz dizimi:

```
for (i = başl.değeri to|downto bitişDeğeri [by AdımBüyüklüğü] ,  
      statements;  
) ;
```

Burada **başl.değeri** ve **bitişDeğeri** pozitif veya negatif gerçel sayı olabilir. Fakat **AdımBüyüklüğü** değeri pozitif gerçel sayı olmalıdır.

► Example:

```
scalar count;  
scalar globmin / inf /;  
for( count = 1 to 1000,  
    x.l(j) = uniform(x.lo(j), x.up(j));  
    solve ml using nlp min obj;  
    if ( obj.l le globmin, globmin = obj.l; );  
);
```

Text Özellikleri ve Çıktı Formatı

Label names and explanatory texts can be accessed via attributes:

- ▶ `ident.ts`: text associated with identifier
- ▶ `element.tl`: label associated with set element
- ▶ `set.te(element)`: text associated with element of set

Local Item Formatting:

- ▶ Syntax for formatting item output: `item:{<>}width:decimals`
- ▶ `{<>}` specifies whether justified left (<), right (>), or centered (<>)
- ▶ `width` is the field width
- ▶ `decimals` is the number of decimals for numeric output
- ▶ each can be omitted, e.g., `x.l::5`

Global Item Formatting:

- ▶ change field justification and width for all items of a type
- ▶ `.lj`, `.nj`, `.sj`, `.tj`, `.lw`, `.nw`, `.sw`, `.tw` attributes of stream identifier
- ▶ see GAMS User's Guide Section 15.10

Cursor Positioning:

- ▶ `put @n;` moves cursor to column `n` of current line

Dynamic Sets

- ▶ dynamic sets allow elements to be **added** or **removed**
- ▶ dynamic sets are usually **domain-checked**, i.e., **subsets**
- ▶ Syntax: `setname(othersetelement) = yes | no` (add/remove single element)
- ▶ Syntax: `setname(subset) = yes | no` (add/remove another subset)
- ▶ Example:

```
set ice / chocolate, strawberry, cherry, vanilla /;  
Parameter demand(ice) / chocolate 1000, strawberry 500,  
                        cherry 10, vanilla 100 /;
```

```
set sorbet(ice);  
sorbet(ice) = yes;  
sorbet('chocolate') = no;   sorbet('vanilla') = no;
```

```
Set highdemand(ice)  ice creams with high demand;  
highdemand(ice) = (demand(ice) >= 500);
```

- ▶ most often used as **controlling index** in an assignment or equation definition

```
Scalar sumhighdemand;  
sumhighdemand = sum(highdemand, demand(highdemand));
```

Examples: refer to first/last period of discrete-time models

```
Set t / 1*24 /;
Sets tb(t)  base period
      tn(t)  non-base periods
      tt(t)  terminal period;
tb(t) = (ord(t) = 1);
tn(t) = (ord(t) > 1);
tt(t) = (ord(t) = card(t));
Variables level(t), inflow(t), outflow(t);
Equations balance(t)      couple fill levels of reservoir over time
          basebalance(t) define fill level for base period;
* only for time periods > base period
balance(tn).. level(t) =e= level(t-1) + inflow(t) - outflow(t);
* only for base period
basebalance(tb).. level(tb) =e= 100 + inflow(tb) - outflow(tb);
* lower bound on fill level in terminal period
level.lo(tt) = 100;

Alternatively (but less readable):
equation basebalance; basebalance..
sum(tb, level(tb)) =e= 100 + sum(tb, inflow(tb) - outflow(tb));
```


Multidimensional Sets

- ▶ describing **assignments** (relations) between sets

- ▶ Example:

```
sets c  'countries'  / jamaica, haiti, guyana, brazil /  
      h  'harbors'    / kingston, s-domingo, georgetown, belem /  
set   hc(h, c)  harbor to country relation  
      / kingston.jamaica,    s-domingo.haiti  
        georgetown.guyana,   belem.brazil /;
```

Makinelere İş Atama Problemi

Örneği – Dosyadan Okuma

9

Makinelere iş atanması problemini konu alan 20 iş ve 4 makineden oluşan küçük bir örnek düşünelim. Bu problemde, çizelgelemeden ziyade (yani işlerin makinelerde yapılış sırasını önemsemeden), hangi işin hangi makineden yapılması gerektiğine karar verelim. En büyük tamamlanma zamanını minimize etmeyi amaçlayan problemin varsayımları ve kısıtları aşağıdaki gibi olsun:

- Her iş tam olarak bir makineye atanmalıdır (bir iş birden fazla makineye atanamaz ve her iş mutlaka yapılmak zorundadır).
- Her iş her makinede yapılamaz, işler uygun oldukları makineye atanmalıdır.
- Bir makinedeki işlerin tamamlanma zamanı, o makineye atanan işlerin süreleri toplamından küçük olamaz.

İşlerin yapımı için uygun makinelerin tablosu – $b(i,m)$ – ve işlerin makinelerdeki işlem süreleri – $t(i,m)$ – aşağıdaki tablolarda verilmektedir.

b(i,m) i. işin m. makineye uygunluğu

	1	2	3	4
1	0	0	1	1
2	0	1	1	1
3	1	1	1	1
4	0	0	1	1
5	1	1	1	1
6	0	0	1	1
7	1	0	1	1
8	0	0	1	1
9	1	1	0	0
10	1	1	0	1
11	1	0	0	1
12	1	1	0	0
13	1	1	0	0
14	1	0	1	0
15	0	0	1	1
16	1	0	0	1
17	1	0	1	1
18	0	0	1	1
19	0	1	0	1
20	0	0	1	1

t(i,m) i. işin m. makinedeki süresi

	1	2	3	4
1	7	15	6	4
2	75	60	80	68
3	12	8	24	20
4	80	75	46	48
5	10	8	26	33
6	22	28	6	8
7	40	35	4	6
8	18	23	16	20
9	20	18	30	28
10	9	12	17	15
11	7	9	10	5
12	8	7	11	13
13	72	60	80	85
14	48	55	58	63
15	14	12	23	28
16	8	10	7	5
17	8	10	9	11
18	18	13	16	14
19	7	4	6	5
20	24	26	14	12

GAMS Kodları-1. Yol

W12MakineAtama.gms

```
Option optcr = 0.00;

sets
i iş indisi /1*20/
m makine indisi /1*4/

variables
CT(m)
Cmax
X(i,m)
Z
;

binary variables
X (i,m)
;
```

```
table
b(i,m) i. işin m. makineye uygunluğu

      1      2      3      4
1      0      0      1      1
2      0      1      1      1
3      1      1      1      1
4      0      0      1      1
5      1      1      1      1
6      0      0      1      1
7      1      0      1      1
8      0      0      1      1
9      1      1      0      0
10     1      1      0      1
11     1      0      0      1
12     1      1      0      0
13     1      1      0      0
14     1      0      1      0
15     0      0      1      1
16     1      0      0      1
17     1      0      1      1
18     0      0      1      1
19     0      1      0      1
20     0      0      1      1
;
```

```
table
t(i,m) i. işin m. makinede işlenme süresi

      1      2      3      4
1      7     15      6      4
2     75     60     80     68
3     12      8     24     20
4     80     75     46     48
5     10      8     26     33
6     22     28      6      8
7     40     35      4      6
8     18     23     16     20
9     20     18     30     28
10      9     12     17     15
11      7      9     10      5
12      8      7     11     13
13     72     60     80     85
14     48     55     58     63
15     14     12     23     28
16      8     10      7      5
17      8     10      9     11
18     18     13     16     14
19      7      4      6      5
20     24     26     14     12
;
```

equations

```
amac      amaç fonksiyonu değerinin hesaplanması
kisit1    her iş tam olarak bir makineye atanmalıdır
kisit2    işler sadece uygun makinelere atanabilir
kisit3    her makinede işlerin tamamlanma zamanlarının hesaplanması
kisit4    en büyük tamamlanma zamanının hesaplanması
;

amac.. Z=e=Cmax;
kisit1(i).. sum((m),X(i,m)) =e= 1 ;
kisit2(i,m).. X(i,m) =l= b(i,m);
kisit3(m).. CT(m)=g= sum((i), X(i,m)*t(i,m));
kisit4(m).. Cmax =g= CT(m);

model MakineAtama_MIN_Cmax/all/;
solve MakineAtama_MIN_Cmax using mip minimizing Z;
display x.l,CT.l;
```

GAMS Kodları-2. Yol (Dosyadan Okuma)

W12MakineAtama_Oku.gms

```
Option optcr = 0.00;

sets
i iş indisi /1*20/
m makine indisi /1*4/

variables
CT(m)
Cmax
X(i,m)
Z
;

binary variables
X (i,m)
;

table
b(i,m) i. işin m. makineye uygunluğu
$include W12uygun.txt
;

table
t(i,m) i. işin m. makinede işlenme süresi
$include W12sure.txt
;
```

W12uygun.txt, W12sure.txt

```
equations
amac amaç fonksiyonu değerinin hesaplanması
kisit1 her iş tam olarak bir makineye atanmalıdır
kisit2 işler sadece uygun makinelere atanabilir
kisit3 her makinede işlerin tamamlanma zamanlarının hesaplanması
kisit4 en büyük tamamlanma zamanının hesaplanması
;

amac.. Z=e=Cmax;
kisit1(i).. sum((m),X(i,m)) =e= 1 ;
kisit2(i,m).. X(i,m) =l= b(i,m);
kisit3(m).. CT(m)=g= sum((i), X(i,m)*t(i,m));
kisit4(m).. Cmax =g= CT(m);

model MakineAtama_MIN_Cmax/all/;
solve MakineAtama_MIN_Cmax using mip minimizing Z;
display x.l,CT.l;
```

Çözüm

---- 86 VARIABLE X.L

	1	2	3	4
1				1.000
2				1.000
3		1.000		
4			1.000	
5	1.000			
6			1.000	
7			1.000	
8			1.000	
9		1.000		
10	1.000			
11	1.000			
12		1.000		
13		1.000		
14	1.000			
15			1.000	
16	1.000			
17	1.000			
18				1.000
19		1.000		
20				1.000

---- 86 VARIABLE CT.L

EMM3208 Optimizasyon Teknikleri - Doç. Dr. İbrahim Küçükkoç
1 90.000, 2 97.000, 3 95.000, 4 98.000

```
Mixed integer rounding cuts applied: 6
Gomory fractional cuts applied: 1
MIP status(101): integer optimal solution
Fixing integer variables, and solving final LP...
Tried aggregator 1 time.
LP Presolve eliminated 109 rows and 86 columns.
All rows and columns eliminated.
Presolve time = 0.00 sec.
Fixed MIP status(1): optimal

Proven optimal solution.

MIP Solution:          98.000000    (40 iterations, 0 nodes)
Final Solve:           98.000000    (0 iterations)

Best possible:          98.000000
Absolute gap:           0.000000
Relative gap:           0.000000

--- Restarting execution
--- W12MakineAtama_Oku.gms(85) 0 Mb
--- Reading solution for model MakineAtama_MIN_Cmax
--- Executing after solve: elapsed 0:00:00.402
--- W12MakineAtama_Oku.gms(86) 3 Mb
*** Status: Normal completion
--- Job W12MakineAtama_Oku.gms Stop 04/18/18 19:28:57 elapsed 0
<
```

Close

Open Log

☐ Summary only ☒ Update

Kaynaklar

Bircan H., Kartal Z. , Doğrusal Programlama Tekniği İle Kapasite Planlaması Yaklaşımı Ve Çimento İşletmesinde Bir Uygulaması, C.Ü. İktisadi ve İdari Bilimler Dergisi, Cilt 5 Sayı 1, 2004.

Dhazn Gillig and Bruce A. McCarl, Introduction to GAMS: Summation Notation with GAMS, Department of Agricultural Economics, Texas A&M University (from Appendix A in McCarl and Spreen's text book on <http://ageco.tamu.edu/faculty/mccarl/mccspr/newa1.pdf>)

Gürdoğan N., Üretim Planlamasında Doğrusal Programlama ve Demir Çelik Endüstrisinde Bir Uygulama, Ankara, 1981.

Ignizio J.P. , Cavalier T.M., Linear Programming, Printice Hall, New Jersey, 1994.

Hamblin, J.E., Math for Liberal Studies, Bin Packing and Scheduling, Shippensburg University, <http://webspace.ship.edu/jehamb/mls/MLS%201-8.pptx> Erişim Tarihi 08.03.2018.

Kabak, Ö. Yöneylem Araştırması – 2 Ders Notları, İstanbul Teknik Üniversitesi, İşletme Fakültesi.

Kucukkoc, I., Doğrusal Programlamada Karışım Problemleri, <http://w3.balikesir.edu.tr/~ikucukkoc/dokumanlar/mixture.pdf>, Erişim Tarihi: 15 Ocak 2018.

Ozyazılım, Lineer Programlama Nedir?, <http://www.ozyazilim.com/ozgur/marmara/karar/lineerprogramlama.htm>, Erişim Tarihi: 15 Ocak 2018.

Rosenthal, Richard E. Gams – A User's Guide, Tutorial, Gams Corporation, USA, 2007.

Salkın, Sultan C. Yöneylem Araştırması – 1 Paket Programlar, İstanbul Teknik Üniversitesi, İşletme Fakültesi.

Sarker, R. A., Newton, C. S., Optimization Modelling: A Practical Approach, CRC Press Book, 2008.

Seçme N.Y., Klasik Doğrusal Programlama Ve Bulanık Doğrusal Programlamanın Karşılaştırmalı Bir Analizi: Üretim Planlama Örneği, Kayseri, 2005.

Sivrikaya, Berna T. Yöneylem Araştırması Gams Uygulama Ders Notları, İstanbul Teknik Üniversitesi, İşletme Fakültesi, 2010.

Ulucan A., Yöneylem Araştırması, İşletmecilik Uygulamalı Bilgisayar Destekli Modelleme, Ankara, 2007.