

Optimizasyon Teknikleri

Ders Notu – 12

**MATLAB Optimization Toolbox
ve Excel'de Optimizasyon**

Prof. Dr. Bilal ALATAŞ

OPTİMİZASYON PROBLEMLERİN ÇÖZÜMÜNDE MATLAB UYGULAMALARI

- GRAFİKSEL OPTİMİZASYON
- KISITLAYICISIZ OPTİMİZASYON
- LİNEER OLMAYAN EŞİTSİZLİK KISITLAYICILI OPTİMİZASYON
- TASARIM DEĞİŞKENLERİN ALACAĞI DEĞERLERİN SINIRLANDIRILMASI
- GRADYANTLARIN TEMİN EDİLDİĞİ OPTİMİZASYON
- EŞİTLİK KISITLAYICILI OPTİMİZASYON

Örnek 1:

Aşağıda verilen optimizasyon probleminin çözümünü grafiksel olarak elde ediniz.

$$\min f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 2)^2$$

s.t.

$$h_1(x_1, x_2) = 2x_1 + x_2 = 8$$

$$h_2(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 4)^2 = 4$$

$$g_1(x_1, x_2) = x_1 + x_2 \leq 7$$

$$g_2(x_1, x_2) = x_1 - 0.25x_2^2 \leq 0$$

$$0 \leq x_1 \leq 10 \qquad 0 \leq x_2 \leq 10$$

```
x1=0:0.1:10;
x2=0:0.1:10;
% Optimizasyon fonksiyonlarının değerdendirileceđi X1 ve X2 matrisleri üretilir
[X1 X2] = meshgrid(x1,x2);
% X1 ve X2 değerdlerine karşılık hedef fonksiyonun değerdleri obj_ex1 dosyası çağırılarak elde edilir
f1 = obj_ex1(X1,X2);% amaç fonksiyonu değerdlendirilir
% X1 ve X2 değerdlerine karşılık g1 kısıtlayıcı fonksiyonun değerdleri inecon1 dosyası çağırılarak elde edilir
ineq1 = inecon1(X1,X2);
ineq2 = inecon2(X1,X2);
eq1 = eqcon1(X1,X2);
eq2 = eqcon2(X1,X2);
% ineq1 kısıtlayıcısına ait iso-line'ler bu kısıtlayıcının 7 değeri için aşağıda verilen komut yardımıyla çizilir
[C1,han1] = contour(x1,x2,ineq1,[7,7],'r-');
% iso-line etiketleri aşağıdaki komut yardımıyla elde edilir
clabel(C1,han1);
% g1 ifadesi mouse tıklaması ile uygun bir yere yerleştirilir
gtext('g1');
% Geri kalan kısıtlayıcılar için de benzer işlemler tekrar edilir
[C2,han2] = contour(x1,x2,ineq2,[0,0],'r--');
clabel(C2,han2);
gtext('g2');
[C3,han3] = contour(x1,x2,eq1,[8,8],'b-');
clabel(C3,han3);
gtext('h1');
[C4,han4] = contour(x1,x2,eq2,[4,4],'b--');
clabel(C4,han4);
gtext('h2');
[C,hf] = contour(x1,x2,f1,'g');
clabel(C,hf);
% grafiđin etiketleri aşağıdaki komut yardımıyla yazılır
xlabel('x1 values','FontName','times','FontSize',12, 'FontWeight','bold'); ylabel('x2 values','FontName','times','FontSize',12, 'FontWeight','bold');
grid
hold off
```

m-dosyasında çağrılan fonksiyonlar aşağıda verilmiştir.

- obj_ex1.m
function retval = obj_ex1(X1,X2)
retval = (X1 - 3).*(X1 - 3) + (X2 - 2).*(X2 - 2);
- eqcon1.m
function retval = eqcon1(X1,X2)
retval = 2.0*X1 + X2;
- eqcon2.m
function retval = eqcon2(X1,X2)
retval = (X1 - 1).*(X1 - 1) + (X2 - 4).*(X2 - 4);
- inecon1.m
function retval = inecon1(X1, X2)
retval = X1 + X2;
- inecon2.m
function retval = inecon2(X1,X2)
retval = X1 - 0.25*X2.^2;

- MATLAB optimizasyon toolbox'ı, optimizasyon probleminin büyüklüğüne göre iki farklı yaklaşım uygulamaktadır:
 - Standart algoritma (Medium-Scale)
 - Büyük ölçekli algoritma (Large-Scale)
- Eğer değişken sayısı çok fazla ise büyük ölçekli algoritma kullanılır. Ancak bu algoritmanın kullanımında, bazı parametrelerin (fonksiyonların gradyanları gibi) kullanıcı tarafından verilmesi gerekir. Ancak Standart algoritma Optimizasyon dersinde verilen örneklerin çözümü için yeterli olduğundan Sadece standart algoritma ve ona bağlı komutlar verilecektir. Bu toolbox'da optimizasyon probleminin tipine bağlı olarak kullanılacak hazır fonksiyonlar bulunmakta, ve bu fonksiyonlar yardımıyla optimizasyon işlemi gerçekleştirilmektedir. Belli başlı optimizasyon fonksiyonları ve işlevleri bir sonraki slayttaki tabloda verilmiştir.
- Optimtool komutu ile grafiksel olarak da toolbox kullanılabilir? (ÖDEV)

Tablo : Optimizasyon komutları ve kullanım amaçları.

Fonksiyon	Amacı
fgoalattain	Birden fazla hedef fonksiyonlu optimizasyon
fminbnd	Skalar nonlinear minimizasyon sınırlarla birlikte
fmincon	Kısıtlayıcı fonksiyonlu nonlinear optimizasyon
fminimax	Minimax optimizasyonu
fminsearch,fminunc	Kısıtlayıcı fonksiyonsuz nonlinear minimizasyon
fseminf	Yarı-sonlu programlama
linprog	Lineer programlama
quadprog	Quadratik programlama

Yukarıdaki tabloda verilen her bir komutun kullanımı, komuta verilmesi gereken değerle (girdi bilgileri), komutun alacağı seçenekler (*OPTIONS*) ve çıktı parametrelerin neler olduğu komut ile ilgi yardım dosyasından elde edilebilir.

En genel optimizasyon problemlerinin çözümünde kullanılan *fmincon* komutu ile ilgili bilgi örnek amacıyla aşağıda verilmiştir.

Tablo : fmincon komutunun kullanımı

<pre>x = fmincon(fun,x0,A,b) x = fmincon(fun,x0,A,b,Aeq,beq) x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub) x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon) x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options) [x,fval] = fmincon(...) [x,fval,exitflag] = fmincon(...) [x,fval,exitflag,output] = fmincon(...) [x,fval,exitflag,output,lambda] = fmincon(...) [x,fval,exitflag,output,lambda,grad] = fmincon(...) [x,fval,exitflag,output,lambda,grad,hessian] = fmincon(...)</pre>
--

Yukarıdaki tabloda, *fmincon* komutunun, en yalın kullanımdan en ileri kullanımı ile ilgili 11 biçimi verilmiştir. Burada temel kullanım benzer olmakla birlikte, optimizasyon problemi ile alakalı ne tür bir bilgi girişi yapılacağı ve optimum çözüm elde edildiğinde ne tür sonuçların, komut tanımlanmada eşitliğin sağ tarafında verilen değişkenlere atanacağı tanımlanır. Dolayısıyla komutta ilgili değişkenlerin mutlaka tanımlı olması gerekir.

Tablo : fmincon komutunun parametrelerinin anlamları.

PARAMETRE	ANLAMI
GİRDİ PARAMETRELERİ	
Fun	hedef fonksiyonu içeren m-dosyasının ismi
X0	tasarım değişkenlerinin başlangıç değerleri
A, b	Lineer eşitliksizlik kısıtlayıcı fonksiyonun katsayıları ($A \cdot x \leq b$)
Aeq, beq	Lineer eşitlik kısıtlayıcı fonksiyonun katsayıları $A_{eq} \cdot x = beq$
lb	Dizayn değişkenlerin alt sınırı
ub	Dizayn değişkenlerin üst sınırı
nonlcon	Kısıtlayıcı fonksiyonları içeren fonksiyon
options	Optimizasyon algoritmasını kontrol eden seçenekler
ÇIKTI PARAMETRELERİ	
x	Optimum tasarım değişkenlerinin atandığı vektör
Fval	Optimum noktada hedef fonksiyonun değeri
output	Optimizasyon çözümü hakkında daha detaylı bilgi verir
Lambda	Lagrange çarpanlarının değerleri
Grad	Optimum noktada hedef fonksiyonun gradyanı
hesssian	Optimum noktada hessian matrisinin değerini verir

- Seçilen optimizasyon metodunun kontrolünü sağlamak için *OPTIONS* parametresi ile belirtilen pek çok argüman vardır. İlgili komutun, *MATLAB* tarafından atanan geçerli (default) değerleri yerine, bu argümanlar *OPTIMSET* komut yardımıyla değiştirilerek kullanıcının tanımladığı ve optimizasyon tekniğini kullandığı değerler değiştirilebilir. Bu argümanlar bir sonraki slaytta verilmiştir.

Tablo : OPTIMSET komutu yardımıyla tanımlanan bazı argümanlar.

ARGUMAN	ANLAMI	ALACAGI DEĞERLER
DerivativeCheck	Kullanıcının sağladığı derivative ile sonlu farklarla hesaplanan derivatı karşılaştırır.	'on', 'off'
DiffMaxChange	Sonlu farklar ile derivative hesaplamada değişkenlerin değerlerindeki maksimum değişiklik	
DiffMinChange	Sonlu farklar ile derivative hesaplamada değişkenlerin değerlerindeki minimum değişiklik	
Display	Çözüm adımlarından hangisinin komut satırında gösterileceği belirtilir	'off': herhangi bir çıktı göstermez 'iter': her bir iterasyonda çıktıları gösterir 'final': sadece son çıktıyı gösterir 'notify': optimizasyon yakınsamadığı zamanki çıktıyı gösterir
GradObj	Hedef fonksiyonun gradyanın kullanıcı tarafından verilir verilmediğini belirtir	'on', 'off'
GradConstr	Nonlinear kısıtlayıcı fonksiyonun gradyanın kullanıcı tarafından verilir verilmediğini belirtir	'on', 'off'
Hessian	Hedef fonksiyonun Hessian matrisinin kullanıcı tarafından verilir verilmediğini belirtir	'on', 'off'
MaxFunEvals	Fonksiyon değerlendirilmesinin maksimum sayısı	Pozitif tam sayı
MaxIter	Maksimum iterasyon sayısı	Pozitif tam sayı
Tolcon	Kısıtlayıcı ihlalindeki tolerans	Pozitif bir sayı
TolFun	Hedef fonksiyon için tolerans	Pozitif bir sayı
TolX	Tasarım değişkenleri için tolerans	Pozitif bir sayı

KISITLAYICISIZ OPTİMİZASYON PROBLEMİ

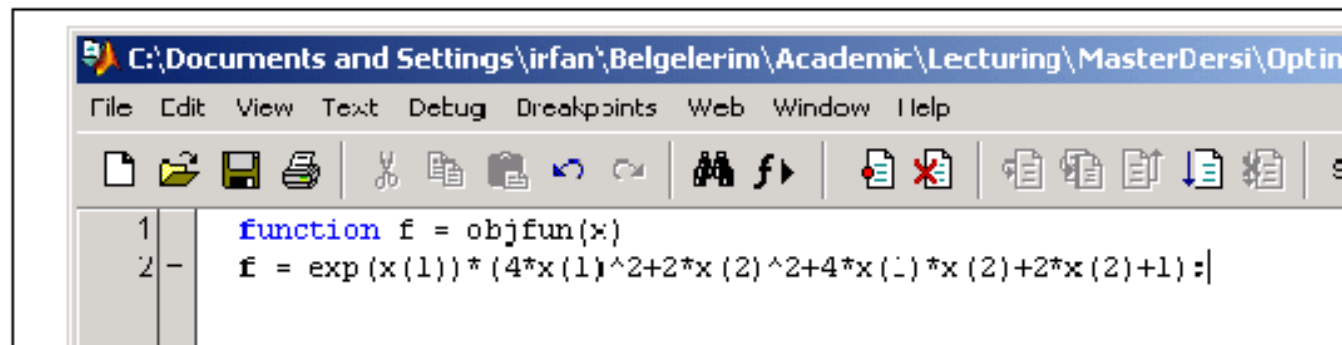
Kısıtlayıcı optimizasyon problemlerinin MATLAB'de çözümü için gerekli adımları aşağıda verilen örnek yardımıyla verilecektir.

$$\underset{x}{\text{minimize}} \quad f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

Bu optimizasyon probleminin çözümü için takip edilecek adımlar:

Adım 1: Bir M-file oluşturun:

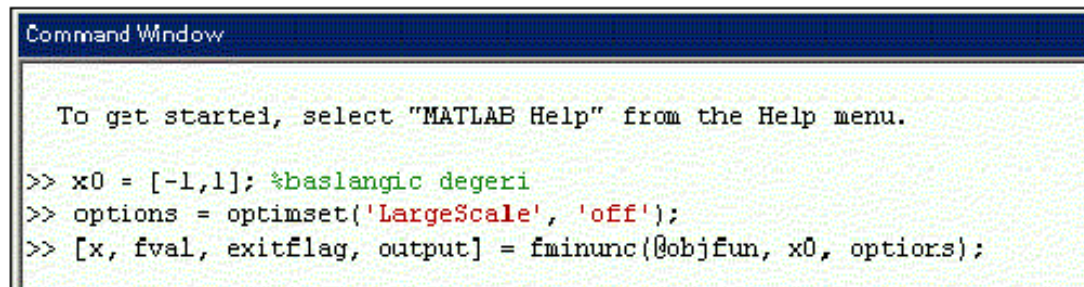
Hedef fonksiyon için bir M-file oluşturulur ve hedef fonksiyona verilen isimle aynı olacak şekilde M-File'in ismi seçilir.



Şekil : objfun.m isimli M-file.

Adım 2: Kısıtlayıcısız optimizasyon metotlarından birini çağırın:

Seçilecek kısıtlayıcısız optimizasyon metoduna göre istenilen opsiyonlarda verilmelidir. Bu örnekte optimizasyon metodu olarak **fminunc** seçilmiştir. Bu komut, tasarım değişkenleri için bir başlangıç değeri atanması gerekmektedir. Gerekli komutlar MATLAB komut satırına aşağıda gösterildiği gibi yazılır:

A screenshot of the MATLAB Command Window. The title bar says "Command Window". The text inside says "To get started, select 'MATLAB Help' from the Help menu." followed by three lines of MATLAB code:

```
>> x0 = [-1,1]; %başlangic degeri
>> options = optimset('LargeScale', 'off');
>> [x, fval, exitflag, output] = fminunc(@objfun, x0, options);
```

Şekil : Optimizasyon komutunun MATLAB komut satırından çağırılması.

Şekil 'de verilen komutların açıklaması:

X0: başlangıç değerleri (tahmini)

Options: optimset komutu ile seçilen optimizasyon seçeneklerinin atandığı değişken. Optimset yardımıyla standart veya büyük ölçekli algoritma seçimine bağlı olarak optimizasyon metodunun seçenekleri belirlenir. Bunlardan bazıları, Jakobien veya Hessian matrisinin hesaplanıp hesaplanmayacağı, ekrana ne kadar bir bilgi yazılacağı gibi seçeneklerdir.

exitflag:Seçilen algoritmanın sonuca yakınsayıp yakınsamadığını belirtir. Eğer sıfırdan büyük ise lokal minimum değerinin bulunduğunu belirtir.

Output:Optimizasyon çözümü hakkında daha detaylı bilgi verir.

örnek için, komut satırına output yazılıp Enter'lanırsa aşağıdaki bilgiler elde edilir:

```
>> output

output =

      iterations: 7
      funcCount: 40
      stepsize: 1
firstorderopt: 8.199771916915478e 004
      algorithm: 'medium-scale: Quasi-Newton line search'
```

Şekil 3: Output komutu yardımıyla optimizasyon çözümü ile ilgili geniş bilgi alma.

Burada:

Iterations: toplam iterasyon sayısını

funcCount: Fonksiyonun değerlendirme sayısı

stepsize: Son iterasyonda seçilen adım uzunluğu

firstorderopt: birinci derece optimumluk şartı

algoritihm: Kullanılan algoritmayı ve seçilen metodu gösterir.

LİNEER OLMAYAN EŞİTSİZLİK KISITLAYICILI OPTİMİZASYON PROBLEMİ

Bu tip optimizasyon problemleri için hedef fonksiyonun yanı sıra eşitsizlik kısıtlayıcıları içinde bir M-file oluşturulur. Yukarıdaki örneğe kısıtlayıcı fonksiyonlar ekleyerek bu bölümde kullanılacak örnek problem oluşturulabilir:

$$\underset{x}{\text{minimize}} \quad f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

kısıtlayıcılar:

$$x_1x_2 - x_1 - x_2 \leq -1.5$$

$$x_1x_2 \geq -10$$

Bu tip optimizasyon problemi için, **fmincon** komutu kullanılarak çözüm yapılabilir.

Adım 1: Kısıtlayıcılar için bir M-dosyasını oluşturulması

```
function [c, ceq] = confun(x)
% Nonlinear inequality constraints
c = [1.5 + x(1)*x(2) - x(1) - x(2);
     -x(1)*x(2) - 10];
% Nonlinear equality constraints
ceq = [];
```

Adım 2: Optimizasyon komutları için bir M-dosyasını oluşturulması

```
x0 = [-1,1]; % Make a starting guess at the solution
options = optimset('LargeScale','off');
[x, fval] = ...
fmincon(@objfun,x0,[],[],[],[],[],[],@confun,options)
```

Buradaki **fmincon** fonksiyonun kullanımlarının genel formatı

```
x = fmincon(@objfun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
```

şeklindedir.

Kısıtlayıcı fonksiyonların çözüm noktasındaki değerlendirmesi için aşağıdaki komut icra edilir:

```
[c,ceq] = confun(x)
```

Bu komutun icrasında elde edilen sonuç:

c =

1.110223024625157e-015

-1.776356839400251e-015

ceq = []

TASARIM DEĞİŞKENLERİN ALACAĞI DEĞERLERİN SINIRLANDIRILMASI

Tasarım değişkenlerin alacağı değerlerde sınırlama olduğu durumda `fmincon` komutu aşağıdaki gibi verilir:

```
x = fmincon(@objfun,x0,[],[],[],[],lb,ub,@confun,options);
```

Burada

`lb`: tasarım değişkenin alacağı alt sınır

`ub`: tasarım değişkenin alacağı üst sınır

Genellikle kullanılan tasarım değişkenlerin sıfırdan büyük değer alma sınırlaması bir önceki problem için aşağıdaki gibi verilebilir:

```
x0 = [-1,1]; % Make a starting guess at the solution
lb = [0,0]; % Set lower bounds
ub = [ ]; % No upper bounds
options = optimset('LargeScale','off');
[x,fval = ...
    fmincon(@objfun,x0,[],[],[],[],lb,ub,@confun,options)
[c, ceq] = confun(x)
```

Bu şartlara göre çözüm yapılırsa aşağıdaki sonuç elde edilir:

```
x =
    0    1.5000
fval =
    8.5000
```

```
c =
    0
   -10
ceq =
    []
```

GRADYANTLARIN TEMİN EDİLDİĞİ OPTİMİZASYON PROBLEMLERİ

- Normalde standart algoritma kullanıldığı durumda gerekli olan gradyantlar sonlu farklar yöntemiyle yaklaşık olarak hesaplanmaktadır. Ancak gradyantların verilebildiği problemlerde gradyantlarında işleme katılması optimizasyon probleminin daha doğru olarak çözülmesini sağlamaktadır.
- Önceki bölümlerde verilen problemin gradyant sağlanarak çözümü için aşağıdaki adımlar takip edilmelidir:

Adım 1: hedef fonksiyon ve gradyant için bir M-file oluşturulur:

```
function [f,G] = objfungrad(x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% Gradient of the objective function
```

```
t = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
G = [ t + exp(x(1)) * (8*x(1) + 4*x(2)),
      exp(x(1))*(4*x(1)+4*x(2)+2)];
```

Adım 2: kısıtlayıcı fonksiyonlar ve gradyantları için bir M-file oluşturulur:

```
function [c,ceq,DC,DCEq] = confungrad(x)
c(1) = 1.5 + x(1) * x(2) - x(1) - x(2); %Inequality constraints
c(2) = -x(1) * x(2)-10;
% Gradient of the constraints
DC= [x(2)-1, -x(2);
      x(1)-1, -x(1)];
% No nonlinear equality constraints
ceq=[];
DCEq = [ ];
```

Gradyantların verildiğinin fmincon 'da belirtmek için optimset komutu yardımıyla Gradobj ve GradConstr değişkenleri 'on' komutunu getirilmelidir:

```
options = optimset(options,'GradObj','on','GradConstr','on');
```

Adım 3: Kısıtlayıcı optimizasyon metotlarından birini çağırın:

```
x0 = [-1,1];           % Starting guess
options = optimset('LargeScale','off');
options = optimset(options,'GradObj','on','GradConstr','on');
lb = [ ]; ub = [ ];    % No upper or lower bounds
[x,fval] = fmincon(@objfungrad,x0,[],[],[],[],lb,ub,...
    @confungrad,options)
[c,ceq] = confungrad(x) % Check the constraint values at x
```

Buradan elde edilen sonuç:

```
x =
    -9.5474    1.0474
fval =
    0.0236
c =
    1.0e-14 *
    0.1110
   -0.1776
ceq =
    []
```

EŞİTLİK KISITLAYICILI OPTİMİZASYON PROBLEMLERİ

Bir önceki örnek için kısıtlayıcılar $x_1^2 + x_2 = 1$ eşitlik kısıtlayıcısı ile $x_1 x_2 \geq -10$ nonlinear eşitsizlik kısıtlayıcıya olacak şekilde seçilsin. Aşağıdaki adımlar icra edilerek optimizasyon problemi çözülür:

Adım 1: Hedef fonksiyon için bir M-dosyasını oluşturulması

```
function f = objfun(x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
```

Adım 2: Kısıtlayıcılar için bir M-dosyasını oluşturulması

```
function [c, ceq] = confuneq(x)
% Nonlinear inequality constraints
c = -x(1)*x(2) - 10;
% Nonlinear equality constraints
ceq = x(1)^2 + x(2) - 1;
```

Adım 3: Kısıtlayıcı optimizasyon metotlarından birini çağırın:

```
x0 = [-1,1]; % Make a starting guess at the solution
options = optimset('LargeScale','off');
[x,fval] = fmincon(@objfun,x0,[],[],[],[],[],[],...
    @confuneq,options)
[c,ceq] = confuneq(x) % Check the constraint values at x
```

Buradan elde edilen sonuç:

```
x =
    -0.7529    0.4332
fval =
    1.5093
c =
   -9.6739
ceq =
    4.0684e-010
```

Excel Solver ile Optimizasyon

Excel 2003 ya da Excel 2007 Kurulumu

Excel Solver (Çözücü) bir MS Excel Eklentisidir ve bir defaya mahsus olmak üzere kurulum gerektirir.

Excel 2003 veya önceki sürümlerde Araçlar Menüsünden eklentilere gelerek yandaki gibi Solver(Çözücü) Eklentisi işaretlenerek seçilmelidir. "Tamam" denildikten sonra Excel Solver (Çözücü) Excelinize yüklenmiş olacaktır.

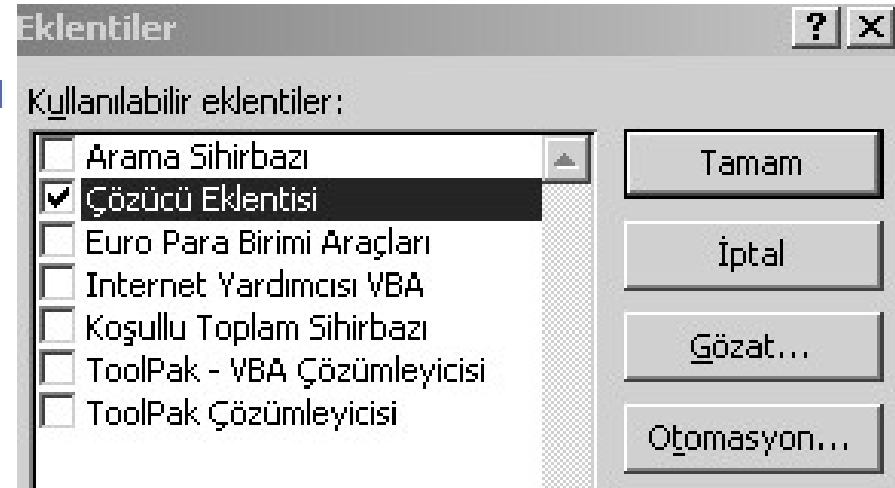
Excel 2007 de kurulum detaylı olarak resimler ile aşağıda gösterilmiştir. Türkçe Excel kullanılıyorsa izlenecek adımlar aynı olup, gerekli açıklamalar ilgili resimlerde belirtilmiştir.

Kurulumu tamamladıktan sonra Solver (Çözücü), 6. resimde olduğu gibi üst menünün Data(Veri) sekmesinde en sağda belirecektir.

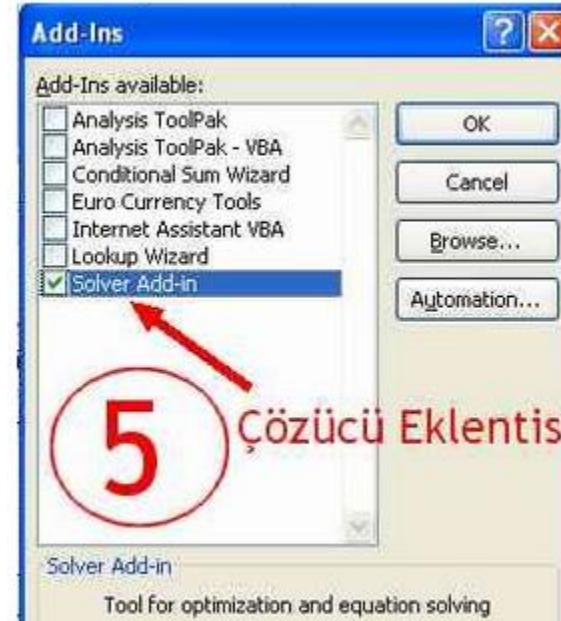
Solver (Çözücü) 'ı çalıştırmak

İlgili Excel Dosyasını açtıktan sonra, Solver (Çözücü) düğmesini (Bkz 6.Resim) tıklayın. Daha sonra 7. resimdeki gibi pencere açılacaktır. Buradan da Solve(Çöz) düğmesine bastığınızda Solver (Çözücü) ilgili çözümü yapacak ve size sunacaktır.

Önemli Not : Solver (çözücü) eklentilerinin çalışması için ilgili Makrolara izin vermeniz gerekmektedir.



Excel Solver ile Optimizasyon



Excel Solver in Doğrusal Programlamada Kullanımı

Excel Solver eklentisini ekledikten sonra bir önceki yazıdaki problemi adım adım çözelim.
Değişkenleri a ve b olarak belirledikten sonra fonksiyonları da aşağıdaki şekilde tekrar belirleyelim.

Problemin amaç fonksiyonu :

$$Z_{\max} = 3a + 2b$$

Problemin Kısıt Fonksiyonları :

$$2a + 1b \leq 100$$

$$1a + 1b \leq 80$$

$$a \leq 40$$

$$a \geq 0, b \geq 0$$

1-İlk olarak boş bir sayfaya uygun girişlerin yapılması gerekmektedir. a değişkeni için A3 hücresi, b değişkeni için de B3 hücresi seçilmiştir. A3 ve B3 hücresini boş bırakıyoruz.

2-Amaç fonksiyonunu C3 hücresine yazıyoruz.
=3*A3+2*B3

3-Kısıt Fonksiyonlarını tanımlama için eşitsizliğin sol tarafı hücrelere formül olarak girilir.

Birinci Kısıt İçin D3 hücresine =2*A3+B3 yazılır

İkinci Kısıt İçin E3 hücresine =A3+B3 yazılır

Üçüncü Kısıt İçin F3 hücresine =A3 yazılır

Dördüncü Kısıt İçin G3 hücresine =A3 yazılır

Beşinci Kısıt İçin H3 hücresine =B3 yazılır

2a + 1b ≤ 100 Eşitsizliği

1a + 1b ≤ 80 Eşitsizliği

a ≤ 40 Eşitsizliği

a ≥ 0 Eşitsizliği

b ≥ 0 Eşitsizliği

4-

Solver Eklenti butonuna bastığımız zaman karşımıza çıkan pencereyi yandaki gibi doldurmamız gerekmektedir.

Yapılması Gerekenler

- Öncelikle Amaç Fonksiyonunu olarak tanımladığımız C3 hücresini resimdeki gibi tanımlamalıyız
- Amaç Ençoklamak (Maksimize Etmek) olduğu için Max olarak belirlememiz gerekiyor.
- Değişken değer olan a ve b için tanımladığımız hücreleri (A3 ve B3) belirtiyoruz.

Solver Parameters

Set Target Cell:

Equal To: ☒ Max ☐ Min ☐ Value of:

By Changing Cells:

Subject to the Constraints:

- \$D\$3 <= 100
- \$E\$3 <= 80
- \$F\$3 <= 40
- \$G\$3 >= 0
- \$H\$3 >= 0

Buttons: Solve, Close, Options, Reset All, Help

- Kısıt Fonksiyonlarını tanımlamak için Ekle butonunu tıklayarak yandaki pencereyi açıyoruz. İlk kısıt fonsiyonumuz $2a + 1b \leq 100$ idi ve D3 hücresinde belirtilmişti. Eşitsizlik Küçük Eşit olduğu için \leq işareti seçilir ve eşitsizliğin sağ tarafındaki sabit değer ilgili alana yazılır.

Change Constraint

Cell Reference:

Constraint:

Buttons: OK, Cancel, Add, Help

- Diğer Kısıt Fonksiyonları da resimlerdeki gibi tanımlanırlar

- Daha sonra Solve (Çöz) butonuna basılarak Excel Solver in optimizasyonu gerçekleştirilmesi istenir. Eğer Solver bir sonuç bulursa otomatik olarak a ve b değişkenlerini A3 ve B3 hücrelerine yazar ve amaç fonksiyonunu hesaplayarak C3 hücresine çıkan sonucu yazar.

Bizim problemimizde

$a = 20$

$b = 60$

$Z_{max} = 180$ olarak bulunmuştur.

Yani A ekmeğinden 20 adet ve B ekmeğinden de 60 adet üretim yapılırsa kar en çoklanmış (maksimize) edilmiş olur. Bu problemde ulaşılabilecek en yüksek kar 180 br liradır ve yukarıda belirtilen kısıtlar altında daha yüksek kar elde edilemez.

En yüksek karı elde etmek için de A ve B ekmeklerinden kaç adet üretileceği cevabı ile beraber problem çözülmüş olur.

Önemli Not : Solver (çözücü) eklentilerinin çalışması için ilgili Makrolara izin vermeniz gerekmektedir.

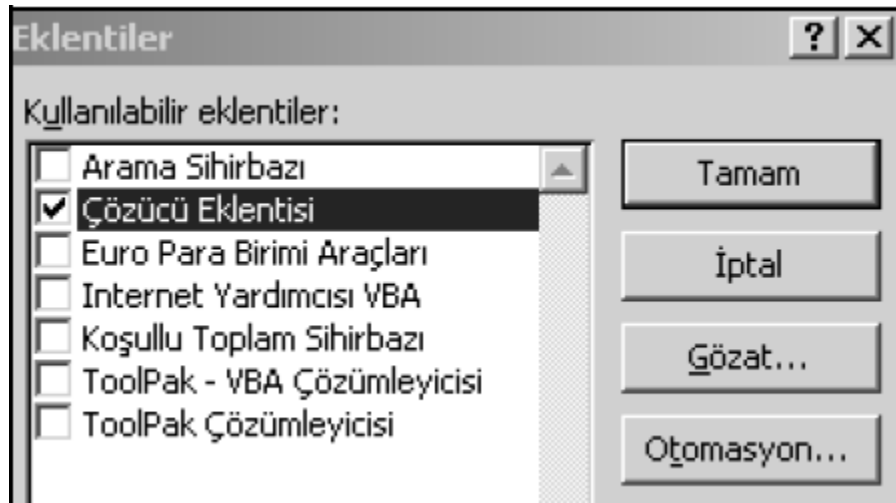
Change Constraint dialog box. Cell Reference: \$E\$3. Constraint: =80. Buttons: OK, Cancel, Add, Help.

Change Constraint dialog box. Cell Reference: \$F\$3. Constraint: =40. Buttons: OK, Cancel, Add, Help.

Change Constraint dialog box. Cell Reference: \$G\$3. Constraint: =0. Buttons: OK, Cancel, Add, Help.

Change Constraint dialog box. Cell Reference: \$H\$3. Constraint: =0. Buttons: OK, Cancel, Add, Help.

Excel ile DP Probleminin çözümü 2. Örnek



BSM

$$Z_{\max} = 5x_1 + 8x_2$$

3.
Hafta

$$4x_1 + 6x_2 \leq 24$$

$$2x_1 + x_2 \leq 18$$

$$3x_1 + 9x_2 \leq 36$$

28.
Sayfa

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

Excel ile DP Probleminin çözümü

Öncelikle Excel çalışma sayfasındaki A1 ve B1 adreslerine X1 ve X2 girilmeli ve A2 ve B2 adreslerine 0 (sıfır) değeri yazılmalıdır (Şekil 2.). Sonraki adımda uygun bir hücreye gelinerek (Örnekte D2 hücresi) bu hücreye amaç fonksiyonu izleyen biçimde yazılır:

$$= 5*A2+8*B2$$

Amaç fonksiyonundan sonra da kısıtlayıcılar benzer şekilde yan hücrelere girilir. Örnek uygulama için kısıtlayıcı fonksiyonların yazılışı ve hücre adresleri izleyen biçimdeki gibi girilmiştir:

E2 Hücresine $=4*A2+6*B2-24$

F2 Hücresine $=2*A2+B2-18$

G2 Hücresine $=3*A2+9*B2-36$

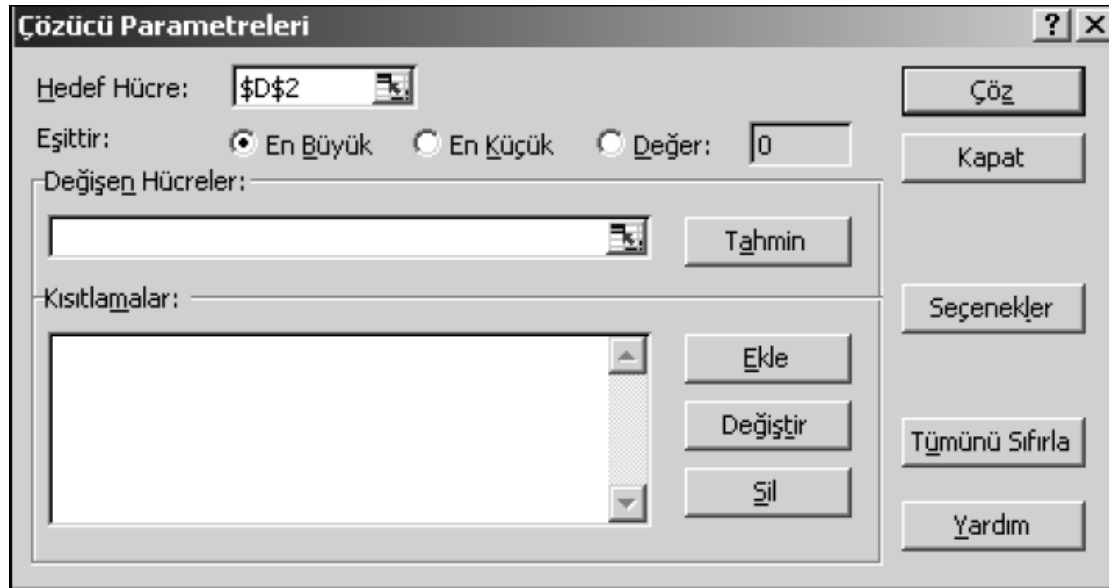
H2 Hücresine $=A2$

I2 Hücresine $=B2$

Bu denklemlerin girilmesinden sonra, çözüm öncesi hücrelerde oluşan durum izleyen şekilde gibidir:

	A	B	C	D	E	F	G	H	I
1	x1	x2							
2	0	0		0	-24	-18	-36	0	0

Excel ile DP Probleminin çözümü



Çözücü Parametreleri

Hedef Hücre:

Eşittir: ☒ En Büyük ☐ En Küçük ☐ Değer:

Değişen Hücreler:

Kısıtlamalar:

Çöz

Kapat

Tahmin

Seçenekler

Ekle

Değiştir

Sil

Tümünü Sıfırla

Yardım

$$Z_{\max} = 5x_1 + 8x_2$$

$$4x_1 + 6x_2 \leq 24$$

$$2x_1 + x_2 \leq 18$$

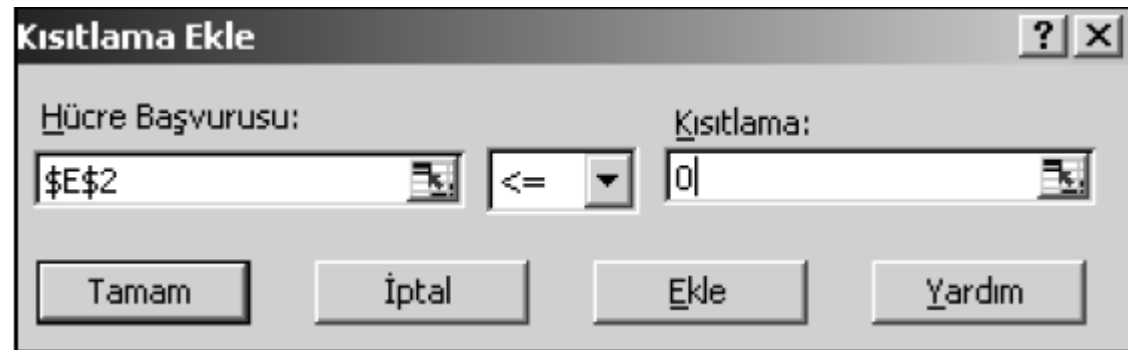
$$3x_1 + 9x_2 \leq 36$$

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

BSM

3.
Hafta

30.
Sayfa



Kısıtlama Ekle

Hücre Başvurusu:

Kısıtlama:

Tamam

İptal

Ekle

Yardım

Excel ile DP Probleminin çözümü

$$Z_{\max} = 5x_1 + 8x_2$$

$$4x_1 + 6x_2 \leq 24$$

$$2x_1 + x_2 \leq 18$$

$$3x_1 + 9x_2 \leq 36$$

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

Çözücü Parametreleri

Hedef Hücre:

Eşittir: ☒ En Büyük ☐ En Küçük ☐ Değer:

Değişen Hücreler:

Kısıtlamalar:

-
-
-
-
-

Butonlar: Çöz, Kapat, Tahmin, Ekle, Değiştir, Sil, Seçenekler, Tümünü Sıfırla, Yardım

BSM

3.
Hafta

	A	B	C	D	E	F	G	H	I
1	x1	x2							
2	0	4		32	0	-14	0	0	4

31.
Sayfa

Kaynaklar

- Doç. Dr. İrfan Kaymaz, MATLAB Optimizasyon Araç Kutusunun Kullanımı, 2006
- <http://www.firattoprak.com/>
- Alan, M.A., Yeşilyurt, C., "Doğrusal Programlama Problemlerinin Excel ile Çözümü", C.Ü.İktisadi ve İdari Bilimler Dergisi, Cilt 5, Sayı 1.