بسم الله الرحمن الرحیم

درس نرم افزارهای ریاضی، آشنایی با نرم‌افزار متلب و لاتک

مدرس: نجمه حسینی منجزی

دانشگاه اصفهان، دانشکده ریاضی و آمار، گروه ریاضیات کاربردی و علوم کامپیوتر

بخش ۹

بهمن ۱۴۰۰

فهرست مطالب

۱ بهینه‌سازی

در حالت کلی یک مسئله بهینه سازی به صورت زیر تعریف می شود

$$\min \quad f(x)$$

$$g(x) \leq 0$$

$$h(x) = 0$$

$$x \in \mathbb{R}^n$$

به طوری که

$$f : \mathbb{R}^n \to \mathbb{R}^{m_1},$$

$$g : \mathbb{R}^n \to \mathbb{R}^{m_2},$$

$$h : \mathbb{R}^n \to \mathbb{R}^{m_3},$$

و $m_1 \in \mathbb{N}$ و $m_2, m_3 \in \mathbb{N} \cup \{0\}$ می باشند. با توجه به اینکه توابع f و g و h چه نوع توابعی باشند و یا اعداد m_1 و m_2 و m_3 مقدار صفر یا غیر صفر داشته باشند مسائل بهینه سازی مختلفی تعریف می شود.

اگر توابع تعریف شده در فوق توابع خطی باشند مسئله یک مسئله بهینه سازی خطی می باشد. این نوع مسائل ساده ترین مسائلی هستند که در بهینه سازی می توان مطالعه کرد. در دوره کارشناسی مسائل از این درس تحت عنوان درس تحقیق در عملیات مطالعه می شوند.

اگر حداقل یکی از توابع استفاده شده در مسئله خطی نباشد مسئله را یک مسئله بهینه‌سازی غیرخطی گویند. برخی از این مسائل در دوره کارشناسی تحت عنوان درس بهینه سازی مطالعه می شوند.

معمولا $m_۱ = ۱$ در نظر گرفته می شود یعنی $f : \mathbb{R}^n \to \mathbb{R}$. دقت کنید اگر $m_۱ > ۱$ آنگاه چند هدف به صورت همزمان داریم و مسئله یک مسئله بهینه‌سازی چندهدفه یا multiobjective می باشد. تابع هدف در این گونه از مسائل به صورت زیر می باشد

$$f(x) = (f_۱(x), f_۲(x), \ldots, f_{m_۱}(x))$$

دقت کنید بعضی از این توابع ممکن است متناقض باشند بنابراین از روش های معمول برای حل این گونه از مسائل نمی توان استفاده کرد و روش های خاص مربوط به خودشان را نیاز دارند. حتی نقاط بهینه در این گونه از مسائل به صورت متفاوت از مسائل معمولی تعریف می شوند.

اگر $m_۲ = m_۳ = ۰$ در نظر گرفته شوند مسئله یک مسئله بهینه‌سازی نامقید نامیده می شود و اگر حداقل یکی از این مقادیر غیرصفر باشد مسئله یک مسئله بهینه‌سازی مقید نامیده می شود.

در این بخش می خواهیم چند دستور از MATLAB برای حل مسائل بهینه‌سازی معرفی کنیم. دستورهایی که در اینجا برای حل مسائل بهینه‌سازی معرفی می کنیم ممکن است از الگوریتم های بهینه‌سازی مختلف پشتیبانی کنند که در اینجا ما به جزئیات این الگوریتم ها اشاره نمی کنیم و تنها به معرفی و نحوه استفاده از دستورات MATLAB اشاره می کنیم. بیان الگوریتم های عددی مختلف بهینه سازی به گذراندن یک درس کاملا مجزا نیاز دارد.

مسائل نامقید :

فرض کنید $f : \mathbb{R}^n \to \mathbb{R}$ یک تابع باشد مسئله زیر یک مسئله بهینه سازی نامقید نامیده می شود:

$$\min \quad f(x)$$

$$x \in \mathbb{R}^n$$

برای حل این گونه از مسائل از تابع fminunc استفاده می کنیم. این تابع min مقدار تابع را بر می گرداند و $x_\circ$ به عنوان نقطه شروع به تابع داده می شود. بنابراین تابع بصورت زیر بکار گرفته می شود

$$>> \quad x = fminunc(fun, x_\circ)$$

$$>> \quad x = fminunc(fun, x_\circ, options)$$

$$>> \quad x = fminunc(problem)$$

$$>> \quad [x, fval] = fminunc(\ldots)$$

$$>> \quad [x, fval, exitflag, output] = fminunc(\ldots)$$

$$>> \quad [x, fval, exitflag, output, grad, hessian] = fminunc(\ldots)$$

برای مثال

```
Command Window
>> fun = @(x)3*x(1)^2 + 2*x(1)*x(2) + x(2)^2 - 4*x(1) + 5*x(2);
>> x0=[1;1];
>> [x,fval] = fminunc(fun,x0)
Warning: Gradient must be provided for trust-region algorithm; using quasi-newton algorithm instead.
> In fminunc (line 397)

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>


x =

    2.2500
   -4.7500


fval =

  -16.3750
```

دقت کنید نقطه اولیه x_0 می تواند به صورت یک بردار افقی یا عمودی تعریف شود

```
Command Window
>> fun = @(x)3*x(1)^2 + 2*x(1)*x(2) + x(2)^2 - 4*x(1) + 5*x(2);
>> x0=[1,1];
>> [x,fval] = fminunc(fun,x0)
Warning: Gradient must be provided for trust-region algorithm; using quasi-n
> In fminunc (line 397)

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>


x =

    2.2500   -4.7500


fval =

  -16.3750
```

و می توانیم با جزییات بیشتر پاسخ را ببینیم و داریم

Command Window

```
>> fun = @(x)x(1)*exp(-(x(1)^2 + x(2)^2)) + (x(1)^2 + x(2)^2)/20;
>> x0=[1,2];
>> [x,fval,exitflag,output] = fminunc(fun,x0)
Warning: Gradient must be provided for trust-region algorithm; using quasi-newton algorithm instead.
> In fminunc (line 397)

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>
```

و

Command Window

```
x =

   -0.6691    0.0000


fval =

   -0.4052


exitflag =

     1


output =

       iterations: 9
        funcCount: 42
         stepsize: 2.9343e-04
     lssteplength: 1
    firstorderopt: 7.9721e-07
        algorithm: 'quasi-newton'
          message: 'Local minimum found....'
```

و اگر بخواهیم مقدار گرادیان و هسین درنقطه بهینه را نیز برای ما نمایش دهد می نویسیم

Command Window

```
>> fun = @(x)x(1)*exp(-(x(1)^2 + x(2)^2)) + (x(1)^2 + x(2)^2)/20;
>> x0=[1,2];
>> [x,fval,exitflag,output,grad,hess] = fminunc(fun,x0)
Warning: Gradient must be provided for trust-region algorithm; using qua
> In fminunc (line 397)

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>


Computing finite-difference Hessian using objective function.
```

که نتیجه به صورت زیر میشود

Command Window

```
output =

       iterations: 9
        funcCount: 42
         stepsize: 2.9343e-04
     lssteplength: 1
    firstorderopt: 7.9721e-07
        algorithm: 'quasi-newton'
          message: 'Local minimum found....'


grad =

   1.0e-06 *

    0.7972
    0.6817


hess =

    1.8998   -0.0000
   -0.0000    0.9552
```

به همراه برگرداندن پاسخ توسط نرم افزار می توانیم از نرم افزار بخواهیم exitflag برای ما برگرداند که این پارامتر یک عدد را برای ما بر می گرداند که مقدار این عدد علت متوقف شدن الگوریتم را مشخص می کند. در لیست زیر مقادیر ممکن برای این مقدار و علت متوقف شدن الگوریتم را در زیر مشخص کرده ایم

exitflag — Reason fminunc stopped
integer

Reason fminunc stopped, returned as an integer.

1	Magnitude of gradient is smaller than the OptimalityTolerance tolerance.
2	Change in x was smaller than the StepTolerance tolerance.
3	Change in the objective function value was less than the FunctionTolerance tolerance.
5	Predicted decrease in the objective function was less than the FunctionTolerance tolerance.
0	Number of iterations exceeded MaxIterations or number of function evaluations exceeded MaxFunctionEvaluations.
-1	Algorithm was terminated by the output function.
-3	Objective function at current iteration went below ObjectiveLimit.

از طرفی با استفاده از options می توانیم برای حل مسئله خصوصی سازی هایی در نظر بگیریم. مثلا می توانیم از برنامه بخواهیم که نتایج حاصل از هر تکرار از الگوریتم را برای ما نمایش دهد و نه صرفا فقط جواب بهینه را. و یا می توانیم نوع الگوریتمی که برای حل مسئله استفاده شود را تعیین کنیم. برای این امر لازم است با الگوریتم های مختلفی که MATLAB استفاده می کند آشنا باشیم و این موضوعات فراتر از این درس می باشند. بنابراین تنها با گفتن دو مورد مربوط به options این بخش را به پایان می بریم.

Command Window

```
>> fun = @(x)x(1)*exp(-(x(1)^2 + x(2)^2)) + (x(1)^2 + x(2)^2)/20;
>> options = optimoptions(@fminunc,'Display','iter','Algorithm','quasi-newton');
>> [x,fval,exitflag,output] = fminunc(fun,x0,options)
                                                        First-order
 Iteration  Func-count       f(x)        Step-size       optimality
     0           3         0.256738                          0.173
     1           6         0.222149             1            0.131
     2           9          0.15717             1            0.158
     3          18        -0.227902      0.438133            0.386
     4          21        -0.299271             1             0.46
     5          30        -0.404028      0.102071           0.0458
     6          33        -0.404868             1           0.0296
     7          36        -0.405236             1          0.00119
     8          39        -0.405237             1         0.000252
     9          42        -0.405237             1         7.97e-07

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>
```

و

```
Command Window

x =

   -0.6691    0.0000


fval =

   -0.4052


exitflag =

     1


output =

       iterations: 9
        funcCount: 42
         stepsize: 2.9343e-04
     lssteplength: 1
    firstorderopt: 7.9721e-07
        algorithm: 'quasi-newton'
          message: 'Local minimum found....'
```

اگر بخواهیم مقدار مینیمم مسئله را بدون استفاده از مشتق محاسبه کند و در واقع با استفاده از روش های derivative free استفاده کند آنگاه دستور fminsearch را بکار می بریم. که همانند تابع معرفی شده در قبل در اینجا نیز می توان این تابع را با ورودی ها و خروجی

های متفاوت بکار برد.

$$>> \quad x = fminsearch(fun, x_{\circ})$$

$$>> \quad x = fminsearch(fun, x_{\circ}, options)$$

$$>> \quad x = fminsearch(problem)$$

$$>> \quad [x, fval] = fminsearch(\ldots)$$

$$>> \quad [x, fval, exitflag, output] = fminsearch(\ldots)$$

در اینجا exitflag می تواند مقادیر زیر را داشته باشد

`exitflag`	Integer identifying the reason the algorithm terminated. The following lists the values of `exitflag` and the corresponding reasons the algorithm terminated.	
	`1`	The function converged to a solution `x`.
	`0`	Number of iterations exceeded `options.MaxIter` or number of function evaluations exceeded `options.MaxFunEvals`.
	`-1`	The algorithm was terminated by the output function.

و مثال زیر را داریم

```
Command Window
>> fun = @(x)x(1)*exp(-(x(1)^2 + x(2)^2)) + (x(1)^2 + x(2)^2)/20;
>> [x,fval,exitflag,output] = fminsearch(fun,x0)

x =

   -0.6691   -0.0000


fval =

   -0.4052


exitflag =

     1


output =

    iterations: 51
     funcCount: 97
     algorithm: 'Nelder-Mead simplex direct search'
       message: 'Optimization terminated:...'

fx >>
```

دقت کنید در اینجا چون از روش های بدون مشتق استفاده می کنیم بنابراین نمی توانیم مقدار گرادیان یا هسین را برگرداند.

اگر مسئله بهینه سازی مورد نظرمان به صورت زیر باشد از روش های معرفی شده در فوق نمی توانیم استفاده کنیم

$$\min \quad f(x)$$

$$x_۱ \leq x \leq x_۲.$$

برای حل این مسئله از دستور fminbnd استفاده می کنیم. که همانند دو تابع بالا به صورت زیر به کار گرفته می شود

$$>> \quad x = fminbnd(fun, x_۱, x_۲)$$

$$>> \quad x = fminbnd(fun, x_۱, x_۲, options)$$

$$>> \quad x = fminbnd(problem)$$

$$>> \quad [x, fval] = fminbnd(\ldots)$$

$$>> \quad [x, fval, exitflag, output] = fminbnd(\ldots)$$

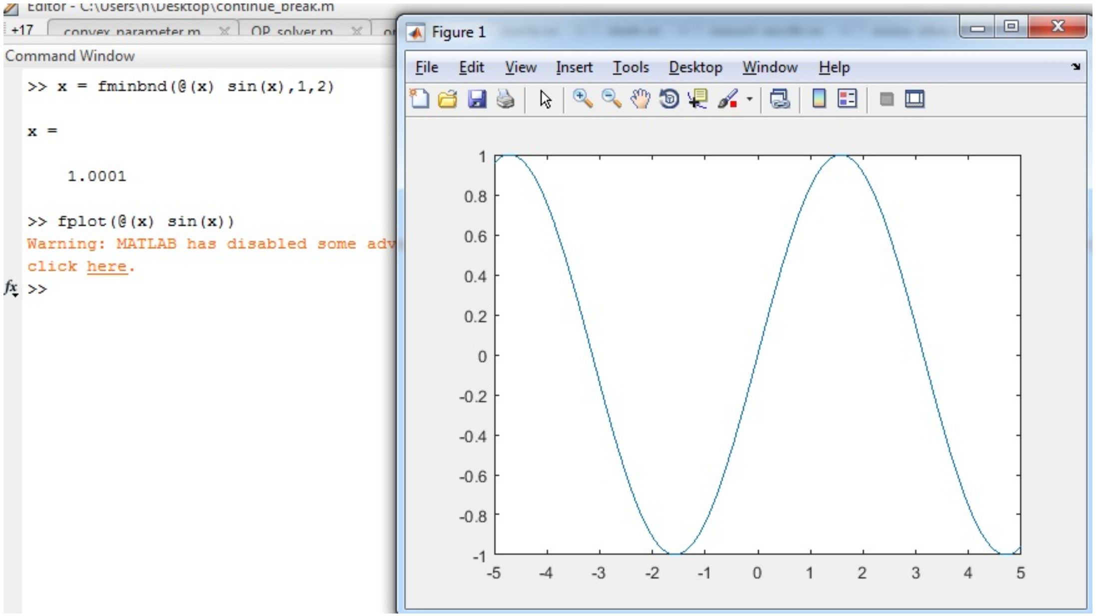
که در اینجا مقادیر ممکن برای exitflag به صورت زیر می باشند

`exitflag`	Integer identifying the reason the algorithm terminated. The following lists the values of `exitflag` and the corresponding reasons the algorithm terminated.	
	1	Function converged to a solution `x`.
	0	Number of iterations exceeded `options.MaxIter` or number of function evaluations exceeded `options.MaxFunEvals`.
	-1	Stopped by an output function or plot function.
	-2	The bounds are inconsistent, meaning `x1 > x2`.

برای مثال داریم

```
Command Window
>> fun = @(x) (x-3)^2-1;
>> [x,fval,exitflag,output] = fminbnd(fun,0,5)

x =

     3


fval =

    -1


exitflag =

     1


output =

    iterations: 5
     funcCount: 6
     algorithm: 'golden section search, parabolic interpolation'
       message: 'Optimization terminated:...'
```

و



مسائل بهینه سازی مقید:

اگر مسئله بصورت یک مسئله بهینه سازی درجه ۲ یا quadratic به فرم زیر باشد از تابع quadprog استفاده می کنیم

$$\min \quad \frac{۱}{۲}x^T H x + f^T x$$

$$Ax \leq b$$

$$(Aeq)x = beq$$

$$lb \leq x \leq ub.$$

که در اینجا H باید یک ماتریس معین مثبت باشد. برای حل این مسئله می نویسیم

$$>> \quad x = quadprog(H, f)$$

$$>> \quad x = quadprog(H, f, A, b)$$

$$>> \quad x = quadprog(H, f, A, b, Aeq, beq)$$

$$>> \quad x = quadprog(H, f, A, b, Aeq, beq, lb, ub)$$

$$>> \quad x = quadprog(H, f, A, b, Aeq, beq, lb, ub, x\circ)$$

$$>> \quad x = quadprog(H, f, A, b, Aeq, beq, lb, ub, x\circ, options)$$

$$>> \quad x = quadprog(...)$$

$$>> \quad [x, fval] = quadprog(H, f, ...)$$

$$>> \quad [x, fval, exitflag] = quadprog(H, f, ...)$$

$$>> \quad [x, fval, exitflag, output] = quadprog(H, f, ...)$$

$$>> \quad [x, fval, exitflag, output, lambda] = quadprog(H, f, ...)$$

که آرگومان های ورودی به صورت زیر می باشند

بهینه‌سازی

Input Arguments

H	Symmetric matrix of doubles. Represents the quadratic in the expression `1/2*x'*H*x + f'*x`.
f	Vector of doubles. Represents the linear term in the expression `1/2*x'*H*x + f'*x`.
A	Matrix of doubles. Represents the linear coefficients in the constraints `A*x ≤ b`.
b	Vector of doubles. Represents the constant vector in the constraints `A*x ≤ b`.
Aeq	Matrix of doubles. Represents the linear coefficients in the constraints `Aeq*x = beq`.
beq	Vector of doubles. Represents the constant vector in the constraints `Aeq*x = beq`.
lb	Vector of doubles. Represents the lower bounds elementwise in `lb ≤ x ≤ ub`.
ub	Vector of doubles. Represents the upper bounds elementwise in `lb ≤ x ≤ ub`.
x0	Vector of doubles. Optional. The initial point for some `quadprog` algorithms: • `active-set` • `trust-region-reflective` when there are only bound constraints If you do not give `x0`, `quadprog` sets all components of `x0` to a point in the interior of the box defined by the bounds. `quadprog` ignores `x0` for the `interior-point-convex` algorithm, and for the `trust-region-reflective` algorithm with equality constraints.

و مقادیری که ممکن است توسط exitflag برگردانده شود به صورت زیر می باشد. البته برای این تابع ممکن است با استفاده از الگوریتم های متفاوت برای حل مسئله چند مقدار دیگر نیز توسط این مقدار برگردانده شود

Integer identifying the reason the algorithm terminated. The following lists the values of `exitflag` and the corresponding reasons the algorithm terminated:

All Algorithms

1	Function converged to the solution x.
0	Number of iterations exceeded `options.MaxIterations`.
-2	Problem is infeasible.
-3	Problem is unbounded.

فرض کنید بخواهیم مسئله بهینه سازی بصورت زیر را حل کنیم

$$\min \quad f(x) = \frac{۱}{۲}x_۱^۲ + x_۲^۲ - x_۱x_۲ - ۲x_۱ - ۶x_۲$$

$$x_۱ + x_۲ \leq ۲$$

$$-x_۱ + ۲x_۲ \leq ۲$$

$$۲x_۱ + x_۲ \leq ۳$$

$$۰ \leq x_۱ \quad ۰ \leq x_۲$$

لازم است ابتدا به فرم ماتریسی نوشته شود بنابراین داریم

$$\mathbf{H} = \begin{pmatrix} ۱ & -۱ \\ -۱ & ۲ \end{pmatrix}, \ \mathbf{f} = \begin{pmatrix} -۲ \\ -۶ \end{pmatrix}, \ \mathbf{x} = \begin{pmatrix} x_۱ \\ x_۲ \end{pmatrix},$$

و قیدها را به صورت زیر تعریف می کنیم

$$\mathbf{A} = \begin{pmatrix} ۱ & ۱ \\ -۱ & ۲ \\ ۲ & ۱ \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} ۲ \\ ۲ \\ ۳ \end{pmatrix}, \ \mathbf{LB} = \begin{pmatrix} ۰ \\ ۰ \end{pmatrix},$$

بنابراین در برنامه مسئله به صورت زیر تعریف می شود

بهینه‌سازی

Command Window

```
>> H = [1 -1; -1 2];
>> f = [-2; -6];
>> A = [1 1; -1 2; 2 1];
>> b = [2; 2; 3];
>> lb = zeros(2,1);
>> [x,fval]=quadprog(H,f,A,b,[],[],[0;0])

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the optimality tolerance,
and constraints are satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


x =

    0.6667
    1.3333


fval =

   -8.2222
```

حال یک مسئله بهینه سازی در فرم کلی به صورت زیر را در نظر بگیرید

$$
\begin{aligned}
\min \quad & f(x) \\
& c(x) \leq 0 \\
& ceq(x) = 0 \\
& Ax \leq b \\
& (Aeq)x = beq \\
& lb \leq x \leq ub.
\end{aligned}
$$

برای حل این مسئله از تابع fmincon استفاده می کنیم. که می توان بصورت زیر آن را بکار برد

$$>> \quad x = fmincon(fun, x_\circ, A, b)$$

$$>> \quad x = fmincon(fun, x_\circ, A, b, Aeq, beq)$$

$$>> \quad x = fmincon(fun, x_\circ, A, b, Aeq, beq, lb, ub)$$

$$>> \quad x = fmincon(fun, x_\circ, A, b, Aeq, beq, lb, ub, nonlcon)$$

$$>> \quad x = fmincon(fun, x_\circ, A, b, Aeq, beq, lb, ub, nonlcon, options)$$

$$>> \quad x = fmincon(...)$$

$$>> \quad [x, fval] = fminconfun, x_\circ, ...)$$

$$>> \quad [x, fval, exitflag] = fmincon(fun, x_\circ, ...)$$

$$>> \quad [x, fval, exitflag, output] = fmincon(fun, x_\circ, ...)$$

$$>> \quad [x, fval, exitflag, output, lambda] = fmincon(fun, x_\circ, ...)$$

$$>> \quad [x, fval, exitflag, output, lambda, grad, hessi] = fmincon(fun, x_\circ, ...)$$

هر کدام از قیدهای فوق را که نداریم به جای آن ماتریس تهی قرار می دهیم. دقت کنید اگر برای چندین متغیر کران پایین داشتیم و برای چند متغیر کران پایین نداشتیم برای مواردی که نداریم کران پایین را منفی بینهایت قرار می دهیم. به طور مشابه برای کران بالا بجای هر کدام

از متغیرها که کران بالا نداریم مثبت بی نهایت قرار می دهیم. در واقع بصورت زیر قرار می دهیم

$$-Inf \qquad Inf$$

برای مثال می خواهیم جواب مسئله زیر را بدست آوریم

$$\min \quad f(x) = 100(x(2) - x(1)^2 + (1 - x_1)^2)$$

$$x_1^2 + x - 2^2 \leq 1$$

که به صورت زیرآن را در نرم افزار پیاده سازی می کنیم

Editor - C:\Users\h\Desktop\my_con.m

+18 | QP_solver.m | oracle1.m | oracle.m | startx.m | inexact_results.m | prime_nhm.m | continue_

```
function [c,ceq]=my_con(x)
c=x(1)^2+x(2)^2-1;
ceq=[];
```

Command Window

```
>> fun=@(x) 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
>> nonlcon=@ my_con;
>> [x,fval]=fmincon(fun,[0;0],[],[],[],[],[],[],nonlcon)

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the optimality tolerance,
and constraints are satisfied to within the default value of the constraint tolerance.

<stopping criteria details>


x =

    0.7864
    0.6177


fval =

    0.0457
```

و می توانیم با جزییات بیشتر ببینیم

Command Window

```
>> fun=@(x) 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
>> nonlcon=@ my_con;
>> options = optimoptions('fmincon','Display','iter','Algorithm','sqp');
>> [x,fval,exitflag,output,lambda,grad,hess]=fmincon(fun,[0;0],[],[],[],[],[],[],nonlco
                                                           Norm of First-order
 Iter F-count            f(x) Feasibility  Steplength        step  optimality
    0       3    1.000000e+00   0.000e+00                              2.000e+00
    1      12    8.913011e-01   0.000e+00   1.176e-01   2.353e-01   1.107e+01
    2      22    8.047847e-01   0.000e+00   8.235e-02   1.900e-01   1.330e+01
    3      28    4.197517e-01   0.000e+00   3.430e-01   1.217e-01   6.153e+00
    4      31    2.733703e-01   0.000e+00   1.000e+00   5.254e-02   4.587e-01
    5      34    2.397111e-01   0.000e+00   1.000e+00   7.498e-02   3.029e+00
    6      37    2.036002e-01   0.000e+00   1.000e+00   5.960e-02   3.019e+00
    7      40    1.164353e-01   0.000e+00   1.000e+00   1.459e-01   1.058e+00
    8      43    1.161753e-01   0.000e+00   1.000e+00   1.754e-01   7.383e+00
    9      46    5.901601e-02   0.000e+00   1.000e+00   1.547e-02   7.278e-01
   10      49    4.533081e-02   2.898e-03   1.000e+00   5.393e-02   1.252e-01
   11      52    4.567454e-02   2.225e-06   1.000e+00   1.492e-03   1.679e-03
   12      55    4.567481e-02   4.406e-12   1.000e+00   2.095e-06   1.501e-05
   13      58    4.567481e-02   0.000e+00   1.000e+00   2.203e-12   1.406e-05

Local minimum possible. Constraints satisfied.
```

و

Command Window

```
x =

    0.7864
    0.6177


fval =

    0.0457


exitflag =

     2


output =

         iterations: 13
          funcCount: 58
          algorithm: 'sqp'
            message: 'Local minimum possible. Constraints satisfied....'
    constrviolation: 0
           stepsize: 2.2030e-12
       lssteplength: 1
      firstorderopt: 1.4058e-05
```

و

Command Window

```
lambda =

         eqlin: [0x1 double]
      eqnonlin: [0x1 double]
       ineqlin: [0x1 double]
         lower: [2x1 double]
         upper: [2x1 double]
    ineqnonlin: 0.1215


grad =

   -0.1911
   -0.1501


hess =

  497.3148 -314.5369
 -314.5369  200.3274
```

در واقع جزییات این تابع به صورت زیر می باشد

Field Name	Entry
`objective`	Objective function
`x0`	Initial point for x
`Aineq`	Matrix for linear inequality constraints
`bineq`	Vector for linear inequality constraints
`Aeq`	Matrix for linear equality constraints
`beq`	Vector for linear equality constraints
`lb`	Vector of lower bounds
`ub`	Vector of upper bounds
`nonlcon`	Nonlinear constraint function
`solver`	`'fmincon'`
`options`	Options created with `optimoptions`

که مقادیر مختلف برای exitflag به صورت زیر می باشد. البته ممکن است برای بعضی الگوریتم ها مقادیر دیگری نیز داشته باشیم و بدلیل اینکه با جزییات این الگوریتم ها آشنا نیستیم به آنها نمی پردازیم

`exitflag` — Reason `fmincon` stopped
integer

Reason `fmincon` stopped, returned as an integer.

All Algorithms:

`1`	First-order optimality measure was less than `options.OptimalityTolerance`, and maximum constraint violation was less than `options.ConstraintTolerance`.
`0`	Number of iterations exceeded `options.MaxIterations` or number of function evaluations exceeded `options.MaxFunctionEvaluations`.
`-1`	Stopped by an output function or plot function.
`-2`	No feasible point was found.

یکی دیگر از مسائلی که ممکن است با آن مواجه شویم مسئله کمترین مربعات می باشد که این مسئله هم در حالت مقید و هم نامقید کاربردهای مختلفی در مسائل کاربردی دارد. این مسئله به صورت زیر تعریف می شود

$$\min \quad \frac{1}{2}\|cx-b\|_2^2$$

$$Ax \leq b$$

$$(Aeq)x = beq$$

$$lb \leq x \leq ub.$$

برای حل این مسئله از تابع lsqlin استفاده می کنیم. که این تابع به صورت زیر تعریف می شود

$$>> \quad x = lsqlin(c, d)$$

$$>> \quad x = lsqlin(c, d, A, b)$$

$$>> \quad x = lsqlin(c, d, A, b, Aeq, beq)$$

$$>> \quad x = lsqlin(c, d, A, b, Aeq, beq, lb, ub)$$

$$>> \quad x = lsqlin(c, d, A, b, Aeq, beq, lb, ub, x\circ)$$

$$>> \quad x = lsqlin(c, d, A, b, Aeq, beq, lb, ub, x\circ, options)$$

$$>> \quad x = lsqlin(...)$$

$$>> \quad [x, resnorm, residual] = lsqlin(c, d, ...)$$

$$>> \quad [x, resnorm, residual, exitflag] = lsqlin(c, d, ...)$$

$$>> \quad [x, resnorm, residual, exitflag, output] = lsqlin(c, d, ...)$$

$$>> \quad [x, resnorm, residual, exitflag, output, lambda] = lsqlin(c, d, ...)$$

که ورودی های این تابع به صورت زیر می باشند

`C`	Matrix multiplier in the term `C*x - d`
`d`	Additive constant in the term `C*x - d`
`Aineq`	Matrix for linear inequality constraints
`bineq`	Vector for linear inequality constraints
`Aeq`	Matrix for linear equality constraints
`beq`	Vector for linear equality constraints
`lb`	Vector of lower bounds
`ub`	Vector of upper bounds
`x0`	Initial point for `x`
`solver`	`'lsqlin'`
`options`	Options created with `optimoptions`

بعلاوه مقادیر exitflag برای این تابع به صورت زیر می باشند

`1`	Function converged to a solution `x`.
`3`	Change in the residual was smaller than the specified tolerance.
`0`	Number of iterations exceeded `options.MaxIterations`.
`-2`	The problem is infeasible.
`-4`	Ill-conditioning prevents further optimization.
`-7`	Magnitude of search direction became too small. No further progress could be made.

یک مسئله برنامه ریزی خطی در حالت کلی به صورت زیر تعریف می شود

$$\begin{aligned} \min \quad & f^T x \\ & Ax \leq b \\ & (Aeq)x = beq \\ & lb \leq x \leq ub. \end{aligned}$$

برای حل این مسئله از تابع linprog استفاده می کنیم. که این تابع به صورت زیر تعریف می شود

$$>> \quad x = linprog(f, A, b)$$

$$>> \quad x = linprog(f, A, b, Aeq, beq)$$

$$>> \quad x = linprog(f, A, b, Aeq, beq, lb, ub)$$

$$>> \quad x = linprog(f, A, b, Aeq, beq, lb, ub, x\circ)$$

$$>> \quad x = linprog(f, A, b, Aeq, beq, lb, ub, x\circ, options)$$

$$>> \quad x = linprog(...)$$

$$>> \quad [x, fval] = linprog(f, A, b, ...)$$

$$>> \quad [x, fval, exitflag] = linprog(f, A, b, ...)$$

$$>> \quad [x, fval, exitflag, output] = linprog(f, A, b, ...)$$

$$>> \quad [x, fval, exitflag, output, lambda] = linprog(f, A, b, ...)$$

برای مثال داریم

Command Window

```
>> A = [1 1;1 1/4; 1 -1; -1/4 -1; -1 -1; -1 1]

A =

    1.0000    1.0000
    1.0000    0.2500
    1.0000   -1.0000
   -0.2500   -1.0000
   -1.0000   -1.0000
   -1.0000    1.0000

>> b = [2 1 2 1 -1 2];
>> Aeq = [1 1/4];
>> beq = 1/2;
>> f = [-1 -1/3];
>> lb = [-1,-0.5];
>> ub = [1.5,1.25];
>> x = linprog(f,A,b,Aeq,beq,lb,ub)
Optimization terminated.

x =

    0.1875
    1.2500

fx >>
```

برای این تابع مقادیر ممکن برای exitflag به صورت زیر می باشد

1	Function converged to a solution x.
0	Number of iterations exceeded options.MaxIterations.
-2	No feasible point was found.
-3	Problem is unbounded.
-4	NaN value was encountered during execution of the algorithm.
-5	Both primal and dual problems are infeasible.
-7	Search direction became too small. No further progress could be made.

برای حل یک مسئله بهینه‌سازی خطی عدد صحیح توام تابع intlinprog در متلب تعریف شده است. این تابع را می‌توان برای یک مسئله به فرم زیر بکار برد:

$$\begin{aligned} \min \quad & f^T x \\ & x(intcon) \text{ integer. is} \\ & Ax \leq b \\ & (Aeq)x = beq \\ & lb \leq x \leq ub. \end{aligned}$$

که این تابع به صورت زیر در متلب استفاده می‌شود:

$$>> \quad x = intlinprog(f, intcon, A, b)$$

$$>> \quad x = intlinprog(f, intcon, A, b, Aeq, beq)$$

$$>> \quad x = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub)$$

$$>> \quad x = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub, x\circ)$$

$$>> \quad x = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub, x\circ, options)$$

$$>> \quad x = linprog(...)$$

$$>> \quad [x, fval] = linprog(f, intcon, A, b, ...)$$

$$>> \quad [x, fval, exitflag] = linprog(f, intcon, A, b, ...)$$

$$>> \quad [x, fval, exitflag, output] = linprog(f, intcon, A, b, ...)$$

برای مثال داریم

$$\begin{aligned} \min \quad & ۴x_۱ + x_۲ \\ & x_۲ \text{ is an integer} \\ & x_۱ + ۲x_۲ \geq -۱۰ \\ & -۴x_۱ - x_۲ \leq -۳۳ \\ & ۲x_۱ + x_۲ \leq ۲۰. \end{aligned}$$

که به صورت زیر در متلب کد می‌شود

```
Command Window
>> f=[4,1];
>> intcon=2;
>> A=[-1 -2;-4 -1;2 1];
>> b=[10 ;-33;20];
>> [x,fval] = intlinprog(f,intcon,A,b)
LP:                Optimal objective value is 33.000000.


Optimal solution found.

Intlinprog stopped at the root node because the objective value is witł
options.AbsoluteGapTolerance = 0 (the default value). The intcon variał
options.IntegerTolerance = 1e-05 (the default value).


x =

    8.2500
         0


fval =

    33
```

با استفاده از این تابع می‌توانیم مسائل ۰ و ۱ را نیز حل کنیم. برای این منظور کافی ست متغیری که می‌خواهیم ۰ یا ۱ باشد را به صورت

عددصحیح در نظر بگیریم و بعد کران پایین آن را ۰ و کران بالا را ۱ تعریف کنیم. برای مثال مسئله زیر را در نظر بگیرید

$$\min \quad -3x_1 - 2x_2 - x_3$$

$$x_x \text{ is a binary}$$

$$x_1 + x_2 + x_3 \leq 7$$

$$4x_1 + 2x_2 + x_3 = 12$$

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

که به صورت زیر در متلب کد می‌شود:

Command Window

```
>> f = [-3;-2;-1];
>> intcon = 3;
>> A = [1,1,1];
>> b = 7;
>> Aeq = [4,2,1];
>> beq = 12;
>> lb = zeros(3,1);
>> ub = [Inf;Inf;1]; % enforces x(3) is binary
>> options = optimoptions('intlinprog','Display','off');
>> [x,fval] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,options)

x =

         0
    5.5000
    1.0000


fval =

  -12.0000

>>
```