بهینه سازی با استفاده از توابع بی‌نام:

گاهی اوقات توابع هدف یا محدودیت ها علاوه بر متغیر مستقل پارامترهایی دارد که این پارامترها سه دسته می باشند:

- توابع بی‌نام (ناشناس)
- توابع تودرتو
- متغیرهای جهانی

متغیرهای جهانی سخت و آزاردهنده هستند چونکه آنها اجازه نمی دهند نام های توابع را دوباره استفاده کنیم لذا بهتر است از دو دسته دیگر استفاده کنیم:

برای مثال فرض کنید می خواهیم تابع هزینه زیر را برای مقادیر مختلف a,b,c کمینه کنیم:

$$f(x) = \left(a - bx_1^2 + x_1^{4/3}\right)x_1^2 + x_1x_2 + \left(-c + cx_2^2\right)x_2^2,$$

حل کننده توابع هدفی را قبول می کند که تنها یک متغیر وابسته داشته باشد (در این مورد x) که در ادامه نشان می دهیم این پارامترها چگونه فراهم می‌آیند:

راه حل هایی که در ادامه به توضیح می آید برای مقادیر پارامتری $a = 4,\ b = 2.1,\ \text{and}\ c = 4$ و نزدیک به نقطه شروع $x_0 = [0.5\ 0.5]$

با استفاده از دستور fminunc می باشد.

توابع بی‌نام:

برای ارسال پارامترها با استفاده از توابع بی‌نام

۱- نوشتن ام فایلی به کد زیر :

www.matlabproject.ir

```
function y = parameterfun(x,a,b,c)
y = (a - b*x(1)^2 + x(1)^4/3)*x(1)^2 + x(1)*x(2) + ...
    (-c + c*x(2)^2)*x(2)^2;
```

۲- اختصاص مقادیر به پارامترها و تعریف تابع عملکرد f به عنوان تابع بی‌نام بوسیله فرامین متلب زیر:

```
a = 4; b = 2.1; c = 4; % Assign parameter values
x0 = [0.5,0.5];
f = @(x)parameterfun(x,a,b,c)
```

۳- فراخوانی حل کننده fminunc با تابع بی‌نام:

```
[x,fval] = fminunc(f,x0)
```

و در نهایت با اجرای برنامه خروجی زیر حاصل می‌شود:

```
Warning: Gradient must be provided for trust-region method;
  using line-search method instead.
> In fminunc at 265
Optimization terminated: relative infinity-norm of gradient less than options.TolFun.
x =
   -0.0898    0.7127
fval =
   -1.0316
```

www.bproject.ir