



Задание №9 от упражненията ***по*** ***Практикум по MATLAB***

Минимум на функции. Минимум на функции с ограничения.
Основна задача на линейното и нелинейното програмиране.
Методи.

Име: **Берна Фердун**
Фак. №: **2213811078**
Специалност: **ИКИ**

Проверил:.....
/гл.ас. Д-р Иван Иванов/

Пловдив,
2024г.

ТЕОРИЯ ПО ЗАДАЧАТА

Съществуват два типа оптимизационни задачи:

1) Задачи за безусловна оптимизация

При този тип задачи обикновено се търсят точки от дефиниционната област $D(f)$ на едномерна или многомерна функция $f(x)$, в които функцията има локални екстремуми (максимум или минимум). Ще отбележим, че областта $D(f)$ може да бъде крайна, безкрайна, отворена, затворена и т.н. Този тип задачи могат да се решават по следните начини:

- С методите на математическия анализ, когато функцията $f(x)$ е диференцируема в дадената област и удовлетворява и някои допълнителни условия. По същество задачата се свежда до решаване на нелинейно уравнение от вида:

$$f'(x) = 0$$

за функция на една променлива или нелинейна система от вида:

$$\begin{cases} \frac{df}{dx_1} = 0 \\ \frac{df}{dx_2} = 0 \\ \dots \\ \frac{df}{dx_n} = 0 \end{cases}$$

за функция на много променливи.

Численото решаване на уравнения и системи уравнения става с методите хордите, Нютон и др. По тази причина няма да разглеждаме отделно тук прилагането на тези методи към задачите на безусловната оптимизация;

- С помощта на специални ефективни алгоритми и числени методи (методи на сканиране, златно сечение, Фибоначи, и др.);
- С числени методи – покоординатно спускане, градиентни методи и др.

2) Оптимизационни задачи с ограничения

Този тип задачи са по-сложни и с тях се занимава голям клон на математиката, наречен математическо оптимиране (програмиране). При тези задачи се търси някакъв минимум или максимум (наречен оптимум) на многомерна функция $f(x)$ $f(x_1, x_2, \dots, x_n)$ при изпълнение на определен брой ограничения (неравенства и уравнения), свързващи независимите променливи..

Според вида на функцията и ограниченията, оптимизационните задачи с ограничения се делят основно на следните три типа:

- Задачи на линейното оптимиране (ЛО) – когато функцията и ограниченията са линейни;
- Задачи на нелинейното оптимиране (НЛО)– когато е на лице някаква нелинейност във функцията и/или в ограниченията;
- Задачи на динамичното оптимиране (ДО) – когато алгоритъмът (стратегията) за достигане до даден оптимум е разпределена на етапи и/или зависи от времето.

Стандартни модели на линейното оптимиране

Много голям брой реални задачи от икономиката, инженерните науки, информатиката и други, се свеждат до търсене на някакво приемливо (оптимално в даден аспект) решение на линейна оптимизационна задача. Ето някои типични примерни задачи, които са обект на линейното оптимиране:

- разпределение на производителни мощности с минимална себестойност
- разпределение на производителни мощности с минимално общо време за работата
- производство с максимална печалба
- разкрояване на материали с минимална загуба на материали
- максимално използване на оборудване
- максимално използване на човешки ресурси
- транспортни задачи – разпределяне на m ресурса (наличности) между n потребителя (потребности) с минимални разходи
- теория на игрите и др.

Код на функции, за изпълнение на задачата:

```
function myfun()
% Инициализация на начални стойности
x0 = [0; 0]; % Начална точка за x1 и x2

% Дефиниране на границите
lb = [0; 0]; % Долни граници: x1 >= 0, x2 >= 0
ub = []; % Без горни ограничения

% Настройки за fmincon
options = optimoptions('fmincon', 'Display', 'iter');

% Извикване на оптимизационната функция
[x, fval] = fmincon(@fun, x0, [], [], [], [], lb, ub, @nonlcon, options);

% Показване на резултатите
fprintf('Намерени минимални стойности:\n');
fprintf('x1 = %.4f\n', x(1));
fprintf('x2 = %.4f\n', x(2));
fprintf('f(x1, x2) = %.4f\n', fval);

% Създаване на 3D визуализация
```

```

[X1, X2] = meshgrid(0:0.1:6, 0:0.1:6); % Област за графиката
F = (X1 - 3).^2 + (X2 - 4).^2;

% Нелинейни ограничения
constraint1 = 2*X1.^2 + X2.^2 <= 34;
constraint2 = 2*X1 + 3*X2 <= 18;
valid = constraint1 & constraint2;

% Визуализация на графиката
figure;
surf(X1, X2, F, 'EdgeColor', 'none', 'FaceColor', 'interp'); % 3D повърхност с
зелен цвят
hold on;
% Изобразяване на областта с ограничения
fill3(X1(valid), X2(valid), min(F(:))*ones(size(X1(valid))), 'r', 'FaceAlpha',
0.3);
plot3(x(1), x(2), fval, 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
xlabel('x1'); ylabel('x2'); zlabel('f(x1, x2)');
title('3D визуализация на f(x1, x2) с ограничения');
hold off;
end

```

```

function [c, ceq] = nonlcon(x)
% Дефиниция на нелинейните ограничения
c = [2*x(1)^2 + x(2)^2 - 34; % Ограничение: 2*x1^2 + x2^2 <= 34
     2*x(1) + 3*x(2) - 18]; % Ограничение: 2*x1 + 3*x2 <= 18
ceq = []; % Липсва равенство за тази задача
end

```

```

function f = fun(x)
% Целева функция: f(x1, x2) = (x1 - 3)^2 + (x2 - 4)^2
f = (x(1) - 3)^2 + (x(2) - 4)^2; % Изчислява стойността на функцията
end

```

<stopping criteria details>

Намерени минимални стойности:

$x_1 = 2.9993$

$x_2 = 3.9993$

$f(x_1, x_2) = 0.0000$

3D визуализация на $f(x_1, x_2)$ с ограничения

