ПРИМЕРЫ решения оптимизационных задач линейного программирования

**Производственная задача (неформализованная)**

Корпорация «АВС Полупроводники» производит микропроцессоры и кристаллы памяти. Для их производства требуются полупроводники типа А и В. Затраты материалов и прибыль от продажи приведены в таблице.

|  |  |  |  |
| --- | --- | --- | --- |
| Производимые изделия | Затраты материалов на изготовление микропроцессоров и кристаллов памяти | | Прибыль  от продажи |
| Полупроводник А | Полупроводник B |
| Микропроцессоры  (1 шт.) | 3 | 5 | 25$  за 1 шт |
| Кристаллы памяти (1000 шт.) | 2 | 10 | 20$  за 1000 шт |
| Ограничения на покупку полупроводников | 450 | 1000 |  |

В силу некоторых ограничений на рынке сырья корпорация может купить только 450 единиц полупроводника А и 1000 единиц материала В. Определите, какое количество микропроцессоров и кристаллов памяти должна производить корпорация «АВС Полупроводники», чтобы **максимизировать** свою прибыль.

Формализуем задачу, выделяя целевую функцию, и ограничения типа неравенств (равенств). Введем обозначения для искомых величин:

– количество (шт) производимых микропроцессоров,

– количество (1000 шт) производимых кристаллов памяти.

Целевая функция – прибыль от продажи

.

Ограничения, накладываемые запасами комплектующих на производство изделий

Для решения данной задачи воспользуемся решателем MATLAB

linprog(f,A,b),

который, по умолчанию, отыскивает минимум целевой функции f

поэтому для поиска максимума функции, нужно знак ее коэффициентов изменить на противоположный.

f=[-25 -20];%min by default

A=[3 2;5 10];

b=[450 1000];

lb=[0 0];

Aeq=[ ];

beq=[ ];

ub=[ ];

x= linprog(f,A,b,Aeq,beq,lb,ub)

x =

125.0000

37.5000

Если желаем получить также значение целевой функции, то задаем команду на решение в виде

[x,fval] = linprog(f,A,b,Aeq,beq,lb,ub)

x =

125.0000

37.5000

fval =

-3.8750e+03

Если задать строку опций и включить **options** во входные аргументы решателя, то вывод решения будет сопровождаться сообщением: Найден минимум, который удовлетворяет ограничениям

options = optimoptions('linprog','Algorithm','interior-point');

x = linprog(f,A,b,Aeq,beq,lb,ub,options)

Minimum found that satisfies the constraints.

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

x =

125.0000

37.5000

**Графическое решение задачи**

В задаче с двумя переменными (и только в этом случае) можно использовать метод графического решения задачи

Чтобы построить границы области изменения переменных, преобразуем все неравенства в равенства, пронумеруем их и выразим как функцию

1)

225-1.5\*;

,

100-0.5\*;

3) ,

Целевая функция при различных значениях переменных равна некоторой константе С

4) 25\*+20\*=C,

,

где обозначено С1=С/20.

Направление ***максимального роста*** *целевой функции f* задает её ***вектор градиент***. Градиент – это вектор, компоненты которого являются частными производными функции f по и :

(f) =  ***\*+ \*= \*+\****

в рассматриваемом случае компоненты градиента равны коэффициентам при и в целевой функции, поскольку 25 и 20.

Вектор градиента функции f направленвдоль прямой y = ***перпендикулярно линии функции*** *(перпендикулярно прямой 25\*+20\*=C)*и указывает направление***максимального роста*** ***функции f.***

Построим графики границ для области изменения переменных, вектор градиента и несколько линий целевой функции для разных значений С1.

Составим команды в кодах МАТЛАБА

X1=0:200;

X21=225-1.5\*X1; %выражаем X2 через X1 из 1-го ограничения

X22=100-0.5\*X1; %выражаем X2 через X1 из 2-го ограничения

plot(X1,X21,'b','LineWidth',2), %график первой линии ограничений

hold on, % все остальные графики строить в тех же осях

plot(X1,X22,'m','LineWidth',2),%график второй линии ограничений

plot(X1,X22-X22, 'r', 'LineWidth',2), %график нижней границы Х2 =0

ylim([0,150];%выделение области решений по y

xlim([0 160]); %выделение области решений по x

grid minor, %мелкая сетка

fplot (@(X1) 20\*X1/25, [0 25],'k','LineWidth',2.5);%построение вектора градиента целевой функции отрезком толщины 2.5 п.

gtext(‘grad f’); %подпись вектор градиента f с выбором места вручную

axis equal;%уравниваем масштаб по осям, чтобы градиент был зрительно перпендикулярен линии целевой функции

% Покажем 3 линии целевой функции f по возрастанию ее значений

Val=[50 100 150]; % числовой массив констант для f

f={'f1','f2','f3'}; % массив ячеек с надписями для линий f

for i=1:3;

C1=Val(i);

fplot(@(X1) C1-1.25\*X1,'g'); %график линии f при разныхС

gtext(f{i}); % подпис ь линии f с выбором места вручную

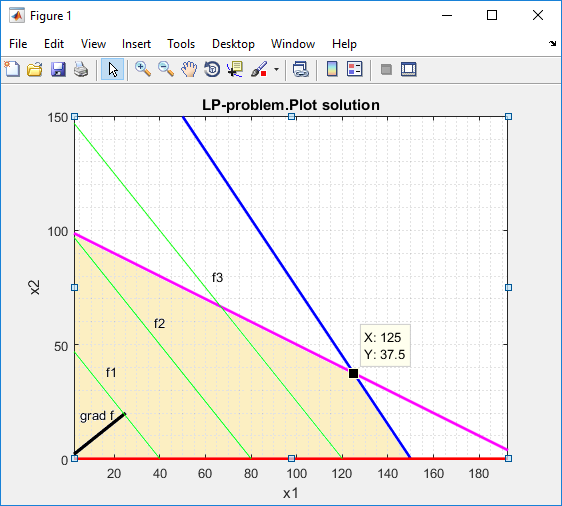
end;

title('LP-problem.Plot solution'); %заголовок графика

xlabel('x1'); %подпись по оси х

ylabel('x2'), %подпись по оси y

hold off



Область допустимых значений переменных и – это окрашенный четырехугольник. Для проверки достаточно подставить в ограничения величины переменных и убедиться, что они выполняются тождественно.

В направлении вектора градиента значения целевой функции возрастают f1<f2<f3, так что она достигает максимального значения на границе области в точке [125, 37.5], которая и является решением:

=125, =37.5.

**Пример (Intrnet) решения формализованной задачи линейного программирования в Matlab**

<http://blablacode.ru/programmirovanie/491>

Найти минимум функции при следующих ограничениях на переменные:

В данном примере есть и равенство, и ограничения нижнего/верхнего значений переменных.  
Сначала зададим матрицу целевой функции:

f = [1; 1; -2];

Теперь зададим матрицу коэффициентов неравенств. При этом все неравенства должны быть приведены к виду "меньше или равно". Если это не так, то меняем знаки.

A = [-5 2 -1; -2 3 -2; 0 0 -1];

Дальше задаем знаки правой части неравенств:

b = [-3; 8; -1];

Задаем нижние ограничения переменных из двойных неравенств:

lb = [1; 2; 1];

Задаем верхние ограничения из двойных неравенств:

ub = [2; 4];

Задаем уравнение:

Aeq = [1 1 1];

beq = 9;

Вычисляем

[x,fval,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub)

Получаем ответ: столбец решений х, значение целевой функции fval, значение показателя выхода из процедуры exitflag, информацию о процессе оптимизации output, а также множители Лагранжа lambda:

Optimization terminated.

x =

1.0000

2.0000

6.0000

fval =

-9.0000

exitflag =

1

output =

iterations: 6

algorithm: 'interior-point'

cgiterations: 0

message: 'Optimization terminated.'

constrviolation: 3.6149e-12

firstorderopt: 7.2028e-11

lambda =

ineqlin: [3x1 double]

eqlin: 2.0000

upper: [3x1 double]

lower: [3x1 double]

Таким образом, найден минимум. Для нахождения максимума целевой функции необходимо сменить знаки в массиве f и повторить расчет.

**Справочная информация из Help MATLAB**

# linprog

Решает задачи линейного программирования

Находит минимум целевой функции при ограничениях

min*xfTx* такой, что

f, x, b, beq, lb, и ub векторы, а A и Aeq матрицы

## Синтаксис

x = linprog(f,A,b)

x = linprog(f,A,b,Aeq,beq)

x = linprog(f,A,b,Aeq,beq,lb,ub)

x = linprog(f,A,b,Aeq,beq,lb,ub,options)

x = linprog(problem)

[x,fval] = linprog(**\_\_\_**)

[x,fval,exitflag,output] = linprog(**\_\_\_**)

[x,fval,exitflag,output,lambda] = linprog(**\_\_\_**)

## Описание

[example](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buus0qo-2)

[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x) = linprog([f](file:///C:\\Program%20Files\\MATLAB\\R2017a\\help\\optim\\ug\\linprog.html" \l "inputarg_f),[A](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_A),[b](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_b)) находит min f'\*x такой, что  A\*x ≤ b.

[example](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buus4rj-1)

[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x) = linprog([f](file:///C:\\Program%20Files\\MATLAB\\R2017a\\help\\optim\\ug\\linprog.html" \l "inputarg_f),[A](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_A),[b](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_b),[Aeq](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_Aeq),[beq](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_beq)) включает ограничения типа равенств Aeq\*x = beq.

Следует ввести пустые матрицы  A = []  и  b = [] , если в условии не имеется неравенств.

[example](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buus4rk-1)

[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x) = linprog([f](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_f),[A](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_A),[b](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_b),[Aeq](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_Aeq),[beq](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_beq),[lb](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_lb),[ub](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_ub)) задает совокупность верхних и нижних границ для искомых величин, x, чтобы решение всегда оставалось в пределах lb ≤ x ≤ ub. Вводим пустые матрицы  Aeq = [] и beq = [] , если в условии не имеется точных равенств.

|  |
| --- |
| **Замечание:**   если указанные границы в задаче противоречивы, то выводится  fval is []. |

[example](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buus4rk-1_1)

[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x) = linprog([f](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_f),[A](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_A),[b](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_b),[Aeq](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_Aeq),[beq](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_beq),[lb](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_lb),[ub](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_ub),[options](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#inputarg_options)) находит минимум согласно указаниям, заданным в options. Для задания options используется команда [optimoptions](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\optimoptions.html) .

[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x) = linprog([problem](file:///C:\\Program%20Files\\MATLAB\\R2017a\\help\\optim\\ug\\linprog.html" \l "inputarg_problem)) отыскивает минимум задачи problem, где problem  - это структура задачи ЛП, описанная в [Input Arguments](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buusznx-2).

Можно создать структуру  problem обращаясь к приложению (app) Optimization, как это описано в [Exporting Your Work](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\graphical-optimization-tool.html#bqu2j29), (ВНИМАНИЕ: в следующих версиях программы этого приложения уже не будет). Можно импортировать структуру problem  из MPS-файла с помощью [mpsread](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\mpsread.html).

[example](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buus4rk-1_2)

[[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x),[fval](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_fval)] = linprog(**\_\_\_**) при любых входных аргументах выводит найденное решение x и значение целевой функции fun  при этом решении: fval = f'\*x.

[example](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buus4rk-1_3)

[[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x),[fval](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_fval),[exitflag](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_exitflag),[output](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_output)] = linprog(**\_\_\_**) дополнительно выводит величину **exitflag** , которая описывает условия выхода из процедуры, а также структуру **output**, в которой содержится информация о процессе оптимизации.

[example](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#buus924-1)

[[x](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_x),[fval](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_fval),[exitflag](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_exitflag),[output](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_output),[lambda](file:///C:\Program%20Files\MATLAB\R2017a\help\optim\ug\linprog.html#outputarg_lambda)] = linprog(**\_\_\_**) дополнительно выводит структуру lambda , в полях которой содержатся множители Лагранжа для найденного решения x.