```
clc;
clear;
close all;
% Group 5:
% Tikhon Riazantsev 382715
% Agastya Heryudhanto

% Exercise sheet 2
% Submission deadline: November 10, 10:00 a.m.
% Task 1: Like Ice in the Sunshine (20 points)
% An ice-cream seller can produce at most 20 kg of ice-cream per hour.
% He sells only two kinds of ice-cream, Chocolate
% Fudge Brownie and Strawberry Cheesecake. His ice-cream machine can
% supply at most 40 kWh per hour. The following table shows further
% particulars of the ice-cream man's business.

%                     | Chocolate Fudge Brownie | Strawberry Cheesecake
%    -----------------------------------------------------------------
%  Cost of production|    35 euro per kg       |    40 euro per kg
%    Retail price    |    60 euro per kg       |    70 euro per kg
%    Needed energy   |     4 kWh per kg        |     5 kWh per kg
%  Marketable amount |      max. 8 kg          |       max. 5 kg

% Although summer has ended, the ice-cream man wants to maximize his
% profits
```

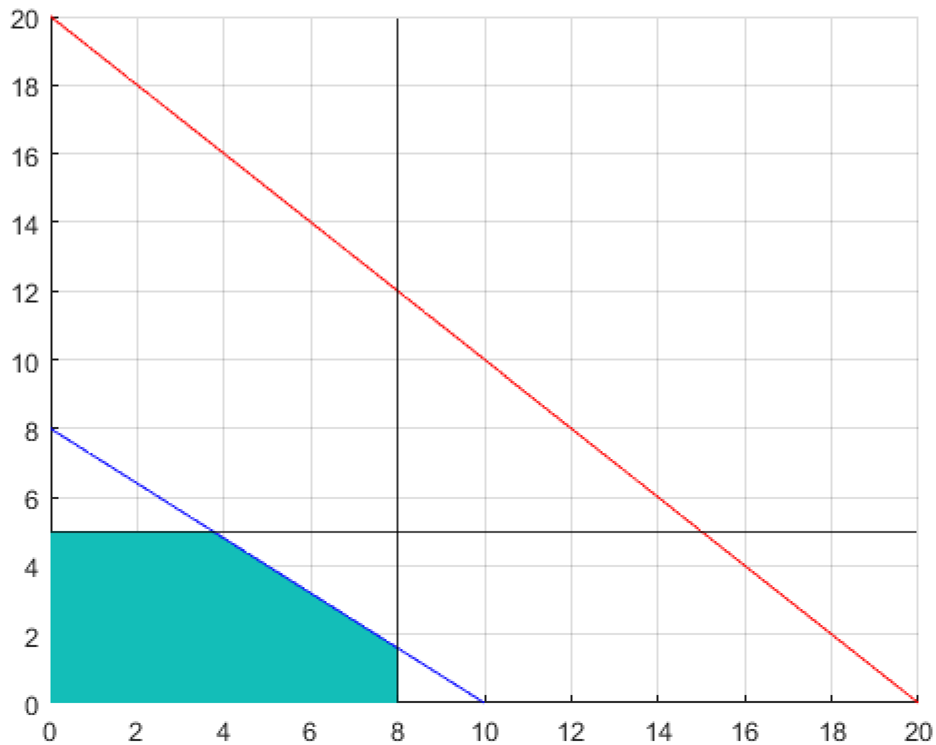
a) Set up the correct Linear Programming problem. Specify all constraints and the objective function. (3 points)

```
% If we consider that:
% x1 - amount of kg of Chocolate Fudge Brownie ice-cream produced;
% x2 - amount of kg of Strawberry Cheesecake ice-cream produced.

% Objective function:
% Maximize: (60-35)*x1 + (70-40)*x2

% Constraints:
% x1 + x2 <= 20
% 4*x1 + 5*x2 <= 40
% x1 + 0*x2 <= 8
% 0*x1 + x2 <= 5
```

b) Draw the solution polyhedron for the given problem. (5 points)

```
% Specifying figure parameters
fig1 = figure;
x1 = (0:0.01:20);
x2 = (0:0.01:20);
hold on;
grid on;
xticks(0:2:20);
yticks(0:2:20);
xlim([0 20]);
ylim([0 20]);
```

```
% Drawing polyhydron with meshgrid and pcolor
[X1,X2] = meshgrid(x1,x2);
constraints = (X1 + X2 <= 20) & (4*X1 + 5*X2 <= 40) & ...
    (X1 + 0*x2 <= 8) & (0*X1 + X2 <= 5);
constraints = double(constraints);
constraints(constraints == 0) = NaN;
h = pcolor(X1,X2,constraints);
h.EdgeColor = 'none';

% Drawing lines of constraints
line1 = fplot(@(x1) 20 - x1, 'r');
line2 = fplot(@(x1) (40 - 4*x1)/5, 'b');
line3 = xline(8,'k');
line4 = yline(5,'k');
```




```
clear;
% c) Find the optimal solution for the given problem by using the
% simplex method. Provide your calculations. (8 points)

% Let us create a simplex tableu from the previous data, knowing from the
% graph that constraint x1 + x2 <= 20 is irrelevant to the solution:

ST = [4 5 1 0 0 40;
      1 0 0 1 0 8;
      0 1 0 0 1 5;
      -25 -30 0 0 0 0];
```

```
% Simplex algorithm (i am so sorry for noodle code but its for speed):
while ~isequal(ST(end,:)<0,zeros(1,size(ST,2)))

    % Finding out entering column and departing row indeces
    entering_column_ind = find(ST(end,:) == min(ST(end,:)));
    departing_row_ind = find(...
        ST(1:end-1,end) ./ ...
        ST(1:end-1,entering_column_ind) == ...
        min(ST(ST(1:end-1,entering_column_ind)>0,end) ./ ...
        ST(ST(1:end-1,entering_column_ind)>0,entering_column_ind)));
    departing_row = ST(departing_row_ind,:);

    % Pivoting matrix
    for i = (1:size(ST,1))
        if ST(i,entering_column_ind) ~= 0
            ST(i,:) = ST(i,:) - ...
                (departing_row * ...
                (ST(i,entering_column_ind) / ...
               departing_row(entering_column_ind)));
        end
    end
    ST(departing_row_ind,:) = departing_row;
end

% Finding the optimised x1 and x2 values from the final simplex tablaeu
% If the coloumns of the tablaeu corresponding to x1 and x2 are basis
% columns, than we can calculate their value. Otherwise the value of the
% non-basis x1 or x2 is zero.

optimal_X = zeros(2,1);
if nnz(ST(:,1)) == 1
    optimal_X(1) = ST(ST(:,1)~=0,end) / ST(ST(:,1)~=0,1);
else
    optimal_X(1) = 0;
end

if nnz(ST(:,2)) == 1
    optimal_X(2) = ST(ST(:,2)~=0,end) / ST(ST(:,2)~=0,2);
else
    optimal_X(2) = 0;
end

% Results:
fprintf("Optimal:\nx1: %d \nx2: %d\n",optimal_X(1),optimal_X(2));
```

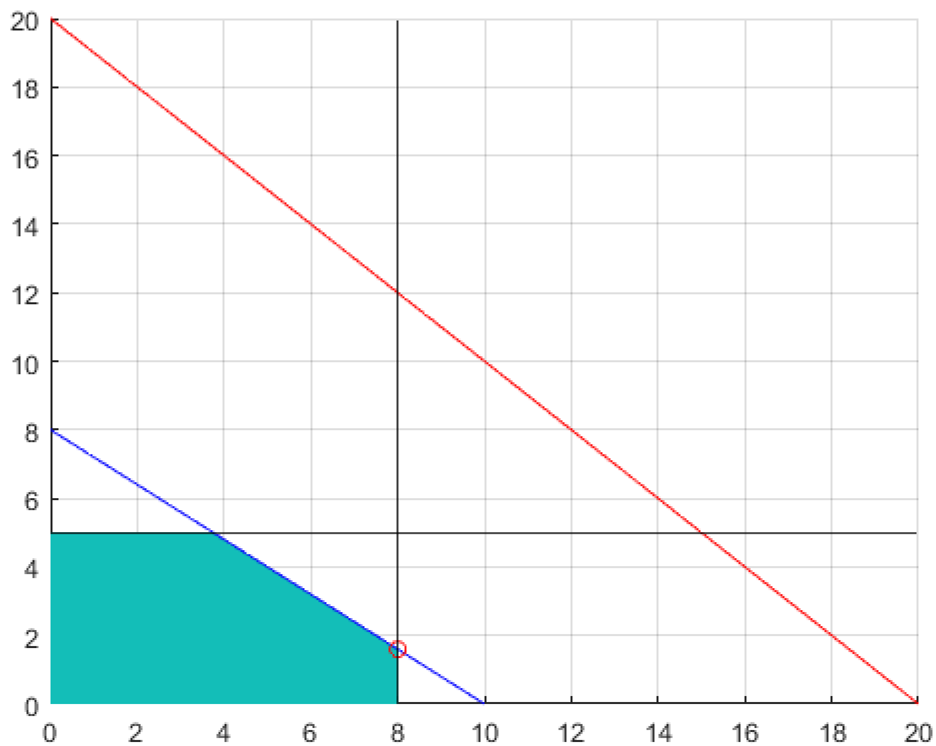
```
Optimal:
x1: 8
x2: 1.600000e+00
```

```
%d)
%Find the optimal solution for the given problem by using the
% linprog function in MATLAB. Provide the commented (!)
% script with the names and student-IDs of all group members in the moodle!
```

```matlab
% (2 points)

% In order to use linprog function, objective, inequality constraints and
% upper and lower bounds has to be defined according to the linprog
% function variables. From simplex solution to vectors.

% Objective Functions coefficients
f = [-25, -30];
% x1 and x2 respectively, negative values because linprog
% default settings finds minimised solutions

% Restating inequality constraints in matrix form
% 4x1 + 5x2 =< 40
% x1 =< 8
% x2 =< 5
A = [4, 5; 1, 0; 0, 1]; % Coefficients of x1 and x2 in the constraints
b = [40; 8; 5]; % Right side coefficients of the matrix


% equality constraints are none, can be defined in function as []

% Definition of the bounds for linprog function variables
Lowerbound = [0; 0]; % Lower bounds
Upperbound = [Inf; Inf]; % no Upper bounds

% Solve Using linprog function
[LinprogSol, fval, exitflag] = linprog(f, A, b, [], [], Lowerbound,
Upperbound);
% LinprogSol is a vector with optimised x1, and x2 as solutions.
% fval calculates the function value (total profit) at the optimised
% solution exitflag is an output that indicates why it exited the function,
% if it does not satisfy feasibilty

% Display the optimal solution in solution polyhedron figure
plot(LinprogSol(1), LinprogSol(2), 'ro');

% Plot the optimal solution as a red dot on polyhedron

% Optimised Solution Using linprog
disp('Optimised Solution:');
disp(['x1 = ', num2str(LinprogSol(1)), ' kg']);
disp(['x2 = ', num2str(LinprogSol(2)), ' kg']);
disp(['Total Profit = ', num2str(-fval), ' euro']);


% e) Describe what a Linear Program and what a Feasibility Test is. What is
% the difference? (2 points)
% Linear program aims to optimise the solutions of a given constraint and
% objective functions. it is for linear problems that have a clear goal to
% minimize or in our case maximise the objective functions by giving the
% optimised solution variables while staying within the defined
% constraints.

% Feasibility tests however, is a method that determines wether the
```

```
% given values for the variables, or a set of variables satisfies the
% constraints of the problem and then checks if the solution is feasible or
% not.

% The main difference between Linear program and feasibility test is that
% linear program optimises solution variables
% for a problem while a Feasibility Test is to check if a
% solution is valid for a given Linear Program.
```

```
Optimal solution found.

Optimised Solution:
x1 = 8 kg
x2 = 1.6 kg
Total Profit = 248 euro
```



Published with MATLAB® R2023b