

---

---

# HW6 ECE 719 Optimal systems

---

---

SPRING 2016  
ELECTRICAL ENGINEERING DEPARTMENT  
UNIVERSITY OF WISCONSIN, MADISON

INSTRUCTOR: PROFESSOR B ROSS BARMISH

BY

NASSER M. ABBASI

DECEMBER 30, 2019

## Contents

0.1	Problem 1 . . . . .	3
0.1.1	Phase one . . . . .	3
0.1.2	Phase 2 . . . . .	6
0.1.3	Verification . . . . .	7
0.2	Problem 2 . . . . .	10
0.2.1	Initial Graphical view . . . . .	10
0.2.2	Solution using Matlab linprog . . . . .	14
0.3	Problem 3 . . . . .	17
0.3.1	Part(a) . . . . .	19
0.3.2	part(b) . . . . .	24

## List of Figures

1	problem 1 description . . . . .	3
2	problem 2 description . . . . .	10
3	Region defined by constraints . . . . .	10
4	Region shown with constraints superimposed . . . . .	11
5	3D plot of the four objective functions . . . . .	11
6	3D plot of the four objective functions with constraint imposed . . . . .	12
7	3D plot of the four objective functions with constraint imposed with optimal solution . . . . .	13
8	problem 3 description . . . . .	18

## List of Tables

## 0.1 Problem 1

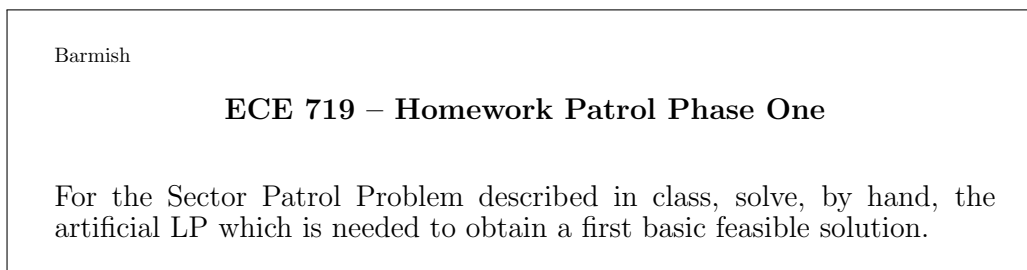


Figure 1: problem 1 description

The patrol problem is given by

$$\begin{aligned} u_i &\geq 0 \\ 2u_1 + 2u_2 &\leq 0 \\ 2u_1 + 2u_2 &\geq 0 \\ u_2 &\geq 1.5u_1 \end{aligned}$$

And the objective function which we want to minimize is

$$J(u) = \frac{1}{30}u_1 + \frac{1}{15}u_2$$

The above is the raw LP. We convert it to standard LP by introducing slack and surplus variable and rename the variables to  $x_i$  from  $u_i$ . Therefore the first table of the initial phase is

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$b$
row 1	2	2	1	0	0	10
row 2	2	2	0	-1	0	4
row 3	-1.5	1	0	0	-1	0
$J(x)$	$\frac{1}{30}$	$\frac{1}{15}$	0	0	0	0

### 0.1.1 Phase one

Since there are two surplus variables (these are the ones associated with  $-1$  entries), we have to start with phase one LP. If there were no surplus variables (i.e. only slack variables), then we go directly to phase two. Phase one is only needed when there are surplus variables. We introduce two new artificial variables  $y_1, y_2$  and an artificial objective  $J(y)$  function which we want to minimize over  $y$  to zero,

$$\min_{y \geq 0, x \geq 0} J(y) = y_1 + y_2$$

The first table of first phase is (where we now use the artificial objective function  $J(y)$  in place of  $J(x)$ .)

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	2	2	1	0	0	0	0	10
row 2	2	2	0	-1	0	1	0	4
row 3	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	0	0	0	0	0	1	1	0

We start by making last row canonical (this means we need to zero out last row entries under  $y_1, y_2$  columns). Doing  $\text{row}(4) = \text{row}(4) - \text{row}(2)$  gives

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	2	2	1	0	0	0	0	10
row 2	2	2	0	-1	0	1	0	4
row 3	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	-2	-2	0	1	0	0	1	-4

To zero out last row under  $y_2$ ,  $\text{row}(4) = \text{row}(4) - \text{row}(3)$  applied to the above gives

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	2	2	1	0	0	0	0	10
row 2	2	2	0	-1	0	1	0	4
row 3	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	-0.5	-3	0	1	1	0	0	-4

We see that the artificial basic feasible solution  $J(y)$  is not optimal, since there are negative values on the last row. second column has the largest negative value in the last row, at  $-3$ . So we need to move this column in the basis vectors. To decide on the pivot row we look at ratio of  $\frac{b}{\text{second column}}$  which gives  $\min\left\{\frac{10}{2}, \frac{4}{2}, \frac{0}{1}\right\} = 0$  which is associated with the third row. So the third row is the pivot row. Now we need to zero out all other entries in the second column. To make (1,2) zero:  $\text{row 1} = \text{row}(1) - (2 \times \text{row}(3))$  gives

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	5	0	1	0	2	0	-2	10
row 2	2	2	0	-1	0	1	0	4
row 3 (pivot)	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	-0.5	-3	0	1	1	0	0	-4

To make (2,2) zero,  $\text{row}(2) = \text{row}(2) - 2 \text{row}(3)$  gives

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	5	0	1	0	2	0	-2	10
row 2	5	0	0	-1	2	1	-2	4
row 3 (pivot)	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	-0.5	-3	0	1	1	0	0	-4

Finally to make (4,2) entry zero,  $\text{row}(4)=\text{row}(4)+3 \text{ row}(3)$  gives

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	5	0	1	0	2	0	-2	10
row 2	5	0	0	-1	2	1	-2	4
row 3 (pivot)	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	-5	0	0	1	-2	0	3	-4

We see that the artificial basic feasible solution is still not optimal since there is negative values on last row. The most negative is in first column. This is the column to move in. Taking the ratio of  $\frac{b}{\text{first column}}$  gives  $\min\left\{\frac{10}{5}, \frac{4}{5}\right\} = \frac{4}{5}$  (we do not divide by negative entries). This minimum is associated with second row. So the second row is the pivot row. We start by normalizing the second row (the pivot row) so that entry (2,1) is one (it is 5 now).  $\text{Row}(2) = \text{row}(2)/5$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	5	0	1	0	2	0	-2	10
row 2 (pivot)	1	0	0	$-\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$-\frac{2}{5}$	$\frac{4}{5}$
row 3	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	-5	0	0	1	-2	0	3	-4

To make (1,1) entry zero, then  $\text{row}(1)=\text{row}(1)-5 \text{ row}(2)$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	0	0	1	1	0	-1	0	6
row 2 (pivot)	1	0	0	$-\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$-\frac{2}{5}$	$\frac{4}{5}$
row 3	-1.5	1	0	0	-1	0	1	0
row 4, $J(y)$	-5	0	0	1	-2	0	3	-4

To make (1,3) zero, then  $\text{row}(3)=\text{row}(3)+1.5 \text{ row}(2)$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	0	0	1	1	0	-1	0	6
row 2 (pivot)	1	0	0	$-\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$-\frac{2}{5}$	$\frac{4}{5}$
row 3	0	1	0	$-\frac{3}{10}$	$-\frac{2}{5}$	$\frac{3}{10}$	$\frac{2}{5}$	$\frac{6}{5}$
row 4, $J(y)$	-5	0	0	1	-2	0	3	-4

To make (4,1) zero, row(4)=row(4)+5 row(2)

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_1$	$y_2$	$b$
row 1	0	0	1	1	0	-1	0	6
row 2 (pivot)	1	0	0	$-\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$-\frac{2}{5}$	$\frac{4}{5}$
row 3	0	1	0	$-\frac{3}{10}$	$-\frac{2}{5}$	$\frac{3}{10}$	$\frac{2}{5}$	$\frac{6}{5}$
row 4, $J(y)$	0	0	0	0	0	1	1	0

We have driven  $J(y)$  to zero with no positive entries in last row. This completes phase one. Now we remove the last row and also remove the  $y_1, y_2$  columns from the above table, and put back the  $J(x)$  in its place in last row. This next tableau is the starting of phase 2.

### 0.1.2 Phase 2

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$b$
row 1	0	0	1	1	0	6
row 2	1	0	0	$-\frac{1}{5}$	$\frac{2}{5}$	$\frac{4}{5}$
row 3	0	1	0	$-\frac{3}{10}$	$-\frac{2}{5}$	$\frac{6}{5}$
$J(x)$	$\frac{1}{30}$	$\frac{1}{15}$	0	0	0	0

We see that all entries in the last row positive, therefore phase two is now complete. There is nothing to do in phase two. All the hard work was done in phase one. This is special case and we were lucky. Note: The text book says that we should now zero out the last row so that zeros appear under the basis columns. But I find this not needed, since it does not change the optimal  $x^*$ . We can always calculate  $J(x)$  once we know  $x^*$ , and there is no need to have  $J(x)$  show up in the bottom right corner of the tableau really. Therefore, I did not do this extra step as not needed.

Now we read out the solution from the above tableau

$$x = \begin{bmatrix} \frac{4}{5} \\ \frac{6}{5} \\ \frac{1}{5} \\ 6 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.2 \\ 0.2 \\ 6 \\ 0 \\ 0 \end{bmatrix}$$

To find the corresponding  $J^*(u)$ , since  $x_1 = u_1$  and  $x_2 = u_2$  and  $J(u) = \frac{1}{30}u_1 + \frac{1}{15}u_2$ , then

$$\begin{aligned} J^* &= \frac{1}{30}(0.8) + \frac{1}{15}(1.2) \\ &= 0.10667 \end{aligned}$$

### 0.1.3 Verification

#### Using Matlab linprog

```

1 function nma_HW6_problem_1
2 %Solves first problem, HW 6, ECE 719
3 %Nasser M. Abbasi
4
5 f=[1/30,1/15,0,0,0];
6 A=[2,2,1,0,0;
7     2,2,0,-1,0;
8     -1.5,1,0,0,-1];
9 b=[10,4,0];
10 [X,FVAL,EXITFLAG,OUTPUT]=linprog(f,[],[],A,b,zeros(size(f)),[])
11
12 end

```

The output from the above is  
Optimization terminated.

```

X =
0.8000
1.2000
6.0000
0.0000
0.0000

```

```

FVAL =
0.1067

```

```

EXITFLAG =
1

```

```

OUTPUT =
iterations: 7
algorithm: 'interior-point-legacy'
cgiterations: 0
message: 'Optimization terminated.'
constrviolation: 8.8818e-16
firstorderopt: 5.0706e-12

```

Using my own `nma_simple.m` which prints all the intermediate tableau and solutions  $x$  during the search

```

1 f=[1/30,1/15,0,0,0];
2 A=[2,2,1,0,0;
3   2,2,0,-1,0;
4   -1.5,1,0,0,-1];
5 b=[10,4,0];
6 nma_simplex(A,b,f,true)

```

The output from the above is

```

>>>>Current tableau [phase one]
  2.0000   2.0000   1.0000   0   0   1.0000   0   0  10.0000
  2.0000   2.0000   0  -1.0000   0   0   1.0000   0   4.0000
 -1.5000   1.0000   0   0  -1.0000   0   0   1.0000   0
  0   0   0   0   0   1.0000   1.0000   1.0000   0
*****
Current tableau [phase one]
  2.0000   2.0000   1.0000   0   0   1.0000   0   0  10.0000
  2.0000   2.0000   0  -1.0000   0   0   1.0000   0   4.0000
 -1.5000   1.0000   0   0  -1.0000   0   0   1.0000   0
 -2.5000  -5.0000  -1.0000   1.0000   1.0000   0   0   0   0

pivot row is 3
current basic feasible solution is
  0
  0
  0
  0
  0
 10
  4
  0
*****
Current tableau [phase one]
  5.0000   0   1.0000   0   2.0000   1.0000   0  -2.0000  10.0000
  5.0000   0   0  -1.0000   2.0000   0   1.0000  -2.0000   4.0000
 -1.5000   1.0000   0   0  -1.0000   0   0   1.0000   0
 -10.0000   0  -1.0000   1.0000  -4.0000   0   0   5.0000   0

pivot row is 2
current basic feasible solution is
  0.8000
  1.2000
  0
  0
  0
  6.0000
  0
  0

```



```

*****
Current tableau [phase one]
  0      0      1.0000      1.0000      0      1.0000      -1.0000      0      6.0000
1.0000      0      0      -0.2000      0.4000      0      0.2000      -0.4000      0.8000
  0      1.0000      0      -0.3000      -0.4000      0      0.3000      0.4000      1.2000
  0      0      -1.0000      -1.0000      0      0      2.0000      1.0000      8.0000

pivot row is 1
current basic feasible solution is
0.8000
1.2000
6.0000
0
0
0
0
0

*****
Current tableau [phase one]
  0      0      1.0000      1.0000      0      1.0000      -1.0000      0      6.0000
1.0000      0      0      -0.2000      0.4000      0      0.2000      -0.4000      0.8000
  0      1.0000      0      -0.3000      -0.4000      0      0.3000      0.4000      1.2000
  0      0      0      0      0      1.0000      1.0000      1.0000      14.0000

*****
Current tableau [phase two]
  0      0      1.0000      1.0000      0      6.0000
1.0000      0      0      -0.2000      0.4000      0.8000
  0      1.0000      0      -0.3000      -0.4000      1.2000
0.0333      0.0667      0      0      0      0

ans =
  0      0      1.0000      1.0000      0      6.0000
1.0000      0      0      -0.2000      0.4000      0.8000
  0      1.0000      0      -0.3000      -0.4000      1.2000
0.0333      0.0667      0      0      0      0

```

Which gives same answer.

## 0.2 Problem 2

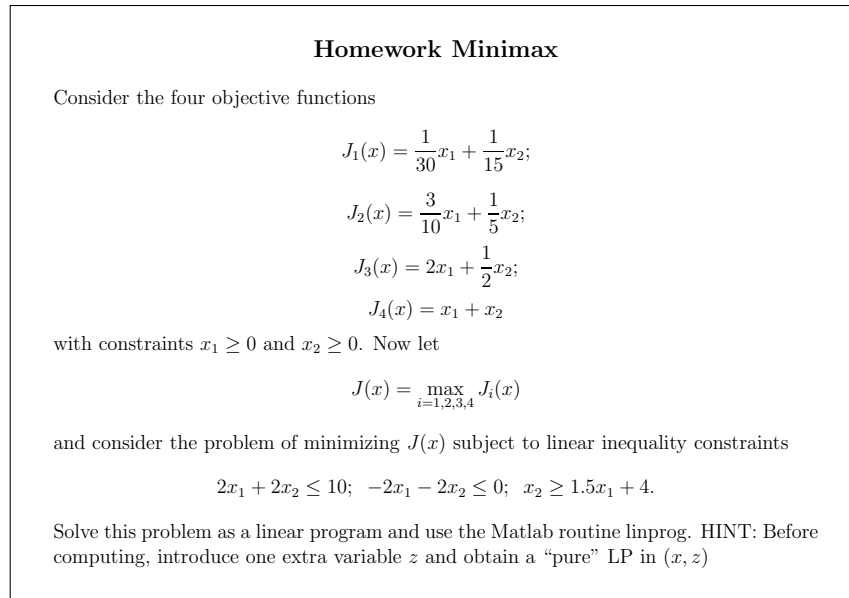


Figure 2: problem 2 description

### 0.2.1 Initial Graphical view

This section shows different views of the problem. In the next section, the solution itself is given. Under each plot, the small code used to generate the plot is shown. First, the feasibility region given by the constraints is plotted.

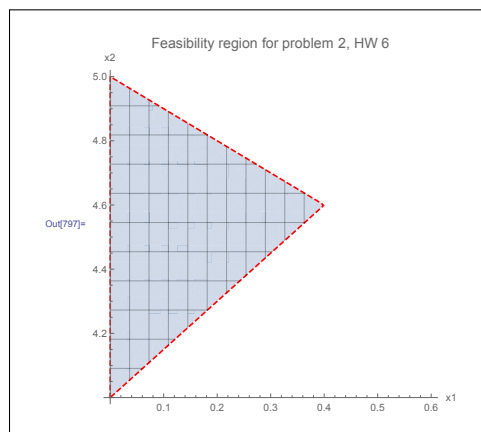


Figure 3: Region defined by constraints

```

1 RegionPlot[
2 2 x1 + 2 x2 <= 10 && -2 x1 - 2 x2 <= 0 && x2 >= 1.5 x1 + 4 && x1 >= 0 &&
3 x2 >= 0, {x1, 0, 0.6}, {x2, 4, 5}, BoundaryStyle -> {Red, Dashed},
4 Mesh -> 10, AxesLabel -> {x1, x2}, Axes -> True, Frame -> None,
5 PlotLabel -> Style["Feasibility region for problem 2, HW 6", 14]]

```

This plot shows each constraint, superimposed on top of the feasibility region.

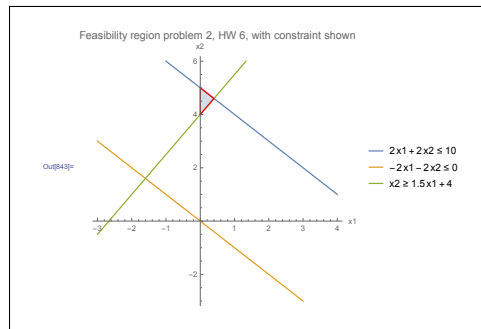


Figure 4: Region shown with constraints superimposed

```

1 p1 = RegionPlot[
2 2 x1 + 2 x2 <= 10 && -2 x1 - 2 x2 <= 0 && x2 >= 1.5 x1 + 4 && x1 >= 0 &&
3 x2 >= 0, {x1, 0, 0.6}, {x2, 4, 5}, BoundaryStyle -> {Red}, Mesh -> None,
4 AxesLabel -> {x1, x2}, Axes -> True, Frame -> None];
5 p2 = ContourPlot[{2 x1 + 2 x2 == 10, -2 x1 - 2 x2 == 0,
6 x2 == 1.5 x1 + 4}, {x1, -3, 4}, {x2, -3, 6},
7 PlotLegends -> {2 x1 + 2 x2 <= 10, -2 x1 - 2 x2 <= 0, x2 >= 1.5 x1 + 4}];
8 Show[p2, p1, AxesLabel -> {x1, x2}, Axes -> True, Frame -> None,
9 PlotRange -> All,
10 PlotLabel ->
11 Style["Feasibility region problem 2, HW 6, with constraint shown", 14]]

```

The following is 3D plot, showing the four objective functions

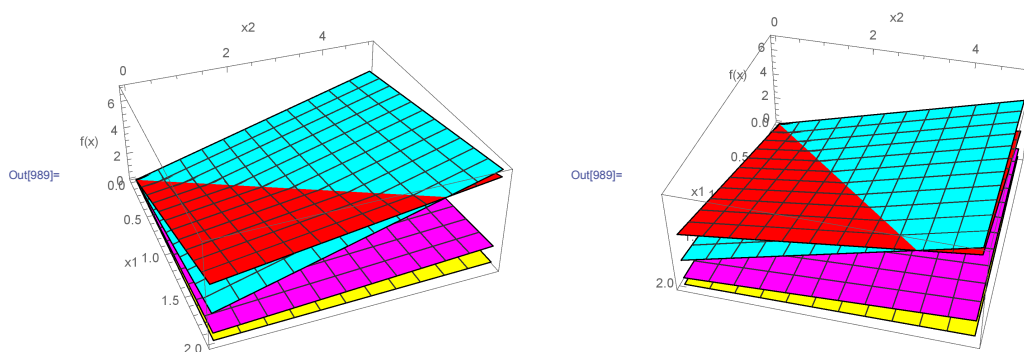


Figure 5: 3D plot of the four objective functions

```

1 funs = {1/30 x1 + 1/15 x2, 3/10 x1 + 1/5 x2, 2 x1 + 1/2 x2, x1 + x2};
2 mesh = 10

```

```

3 g1 = Plot3D[funs[[1]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Yellow,
4 Mesh -> mesh, Lighting -> {"Ambient", White}];
5 g2 = Plot3D[funs[[2]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Magenta,
6 Mesh -> mesh, Lighting -> {"Ambient", White}];
7 g3 = Plot3D[funs[[3]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Red,
8     Mesh -> mesh, Lighting -> {"Ambient", White}];
9 g4 = Plot3D[funs[[4]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Cyan,
10    Mesh -> mesh, Lighting -> {"Ambient", White}];
11 Show[g1, g2, g3, g4, AxesLabel -> {x1, x2, "f(x)"}, PlotRange -> All,
12     SphericalRegion -> True]

```

The following is same 3D plot, but now with the constraints added

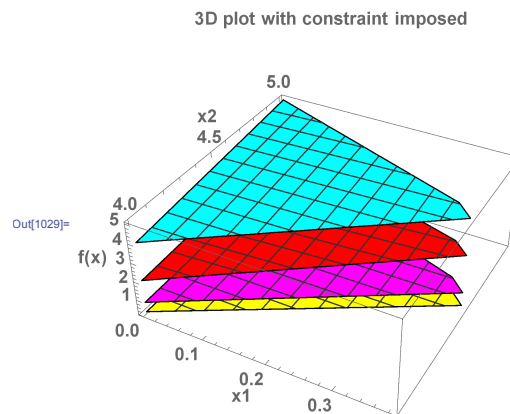


Figure 6: 3D plot of the four objective functions with constraint imposed

```

1 funs = {1/30 x1 + 1/15 x2, 3/10 x1 + 1/5 x2, 2 x1 + 1/2 x2, x1 + x2};
2 mesh = 10;
3 reg = RegionFunction ->
4 Function[{x, y},
5 x >= 0 && y >= 0 && 2 x + 2 y <= 10 &&
6     -2 x - 2 y <= 0 && y >= 1.5 x + 4];
7 g1 = Plot3D[funs[[1]], {x1, 0, 2}, {x2, 0, 5},
8     PlotStyle -> Yellow, Mesh -> mesh,
9     Lighting -> {"Ambient", White}, Evaluate@reg];
10
11 g2 = Plot3D[funs[[2]], {x1, 0, 2}, {x2, 0, 5},
12     PlotStyle -> Magenta, Mesh -> mesh,
13     Lighting -> {"Ambient", White}, Evaluate@reg];
14
15 g3 = Plot3D[funs[[3]], {x1, 0, 2}, {x2, 0, 5},
16     PlotStyle -> Red, Mesh -> mesh,
17     Lighting -> {"Ambient", White}, Evaluate@reg];

```

```

18 g4 = Plot3D[funs[[4]], {x1, 0, 2}, {x2, 0, 5},
19   PlotStyle -> Cyan, Mesh -> mesh,
20   Lighting -> {"Ambient", White}}, Evaluate@reg];
21
22
23 Show[g1, g2, g3, g4, AxesLabel -> {x1, x2, "f(x)"},
24   PlotRange -> All, SphericalRegion -> True,
25   PlotLabel -> Style["3D plot with constraint imposed", 14],
26   BaseStyle -> {Bold, 14}]

```

The solution found is now added the above plot., which is  $x_1 = 0, x_2 = 4$  with the min max value of  $J(u) = 4$  marked with small red point below.

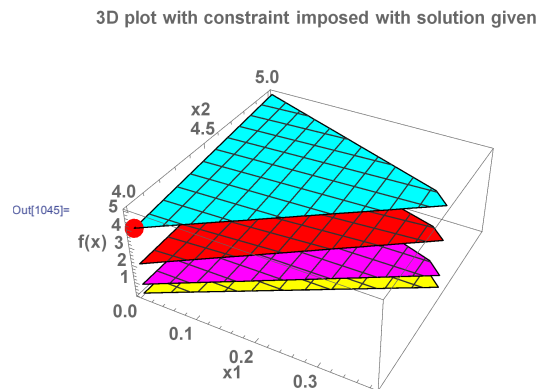


Figure 7: 3D plot of the four objective functions with constraint imposed with optimal solution

```

1 funs = {1/30 x1 + 1/15 x2, 3/10 x1 + 1/5 x2, 2 x1 + 1/2 x2, x1 + x2};
2 mesh = 10;
3 reg = RegionFunction ->
4   Function[{x, y},
5     x >= 0 && y >= 0 && 2 x + 2 y <= 10 && -2 x - 2 y <= 0 && y >= 1.5 x + 4];
6 g1 = Plot3D[funs[[1]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Yellow,
7   Mesh -> mesh, Lighting -> {"Ambient", White}}, Evaluate@reg];
8 g2 = Plot3D[funs[[2]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Magenta,
9   Mesh -> mesh, Lighting -> {"Ambient", White}}, Evaluate@reg];
10 g3 = Plot3D[funs[[3]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Red, Mesh -> mesh,
11   Lighting -> {"Ambient", White}}, Evaluate@reg];
12 g4 = Plot3D[funs[[4]], {x1, 0, 2}, {x2, 0, 5}, PlotStyle -> Cyan,
13   Mesh -> mesh, Lighting -> {"Ambient", White}}, Evaluate@reg];
14 Show[g1, g2, g3, g4,
15   Graphics3D[{Red, PointSize[.05], Point[{0, 4, 4]}], Axes -> True],
16   AxesLabel -> {x1, x2, "f(x)"}, PlotRange -> All, SphericalRegion -> True,
17   PlotLabel ->
18   Style["3D plot with constraint imposed with solution given", 14],

```

19 BaseStyle -> {Bold, 14}]

## 0.2.2 Solution using Matlab linprog

The first step is to convert this multi-objective minimax problem with constraints, to a pure linear programming problem so that Matlab linprog can be used. Introducing extra variable  $z$  the problem can be written as

$$\begin{aligned} \min_z \quad & \tilde{J}(z) = z \\ \text{s.t.} \quad & z \geq J_i(x) \quad i = 1 \dots 4 \\ & 2x_1 + 2x_2 \leq 10 \\ & -2x_1 - 2x_2 \leq 0 \\ & x_2 \geq 1.5x_1 + 4 \end{aligned}$$

Using the  $J_i(x)$  given, the above becomes

$$\begin{aligned} \min_z \quad & \tilde{J}(z) = z \\ \text{s.t.} \quad & \frac{1}{30}x_1 + \frac{1}{15}x_2 - z \leq 0 \\ & \frac{3}{30}x_1 + \frac{1}{5}x_2 - z \leq 0 \\ & 2x_1 + \frac{1}{2}x_2 - z \leq 0 \\ & x_1 + x_2 - z \leq 0 \\ & 2x_1 + 2x_2 \leq 10 \\ & -2x_1 - 2x_2 \leq 0 \\ & x_2 \geq 1.5x_1 + 4 \end{aligned}$$

To use Matlab linprog, we first convert the above to the standard form. We do this by introducing slack and surplus variables. The above becomes (added tilde on top of the  $x$  to make it clear which one variables are the original raw LP variables, and which is ones are the new variables).

$$\begin{aligned} \min_z \quad & \tilde{J}(z) = z \\ \text{s.t.} \quad & \frac{1}{30}x_1 + \frac{1}{15}x_2 - z + \tilde{x}_3 = 0 \\ & \frac{3}{30}x_1 + \frac{1}{5}x_2 - z + \tilde{x}_4 = 0 \\ & 2x_1 + \frac{1}{2}x_2 - z + \tilde{x}_5 = 0 \\ & x_1 + x_2 - z + \tilde{x}_6 = 0 \\ & 2x_1 + 2x_2 + \tilde{x}_7 = 10 \\ & -2x_1 - 2x_2 + \tilde{x}_8 = 0 \\ & -1.5x_1 + x_2 - \tilde{x}_9 = 4 \end{aligned}$$



Below is the Matlab code used to solve the above. The solution is

$$\begin{pmatrix} x_1 \\ x_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \\ \tilde{x}_6 \\ \tilde{x}_7 \\ \tilde{x}_8 \\ \tilde{x}_9 \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 3.7333 \\ 3.2 \\ 2 \\ 0 \\ 2 \\ 8 \\ 0 \\ 4 \end{pmatrix}$$

And minimum of the maximum of  $J_i(x)$  is

$$\min \max J_i(x) = 4$$

```

1 function nma_HW6_problem_2
2 %Solves second problem, HW 6, ECE 719
3 %Nasser M. Abbasi
4
5 f = [0 0 0 0 0 0 0 0 0 1];
6 A = [1/30 1/15 1 0 0 0 0 0 0 0 -1;
7      1/30 1/5 0 1 0 0 0 0 0 0 -1;
8      2 1/2 0 0 1 0 0 0 0 0 -1;
9      1 1 0 0 0 1 0 0 0 0 -1;
10     2 2 0 0 0 0 1 0 0 0 0;
11     -2 -2 0 0 0 0 0 1 0 0 0;
12     -1.5 1 0 0 0 0 0 0 0 -1 0];
13 b=[0 0 0 0 10 0 4];
14
15 [X,FVAL,EXITFLAG,OUTPUT]=linprog(f, [], [], A,b,zeros(size(f)), [])
16 end

```

The output from the above is

```

>> nma_HW6_problem_2
Optimization terminated.
X =
0.0000
4.0000
3.7333
3.2000
2.0000
0.0000

```



```

2.0000
8.0000
0.0000
4.0000
FVAL =
4.0000
EXITFLAG =
1
OUTPUT =
iterations: 7
algorithm: 'interior-point-legacy'
cgiterations: 0
message: 'Optimization terminated.'
constrviolation: 1.4211e-14
firstorderopt: 7.8638e-13

```

This was also solved using Mathematica. Here is the result, which confirms the above as well. This command is from Mathematica help, and used it to apply to this problem:

```

1 SetAttributes[FindMinMax, HoldAll];
2 FindMinMax[{f_Max, cons_}, vars_, opts___?OptionQ] :=
3 With[{res = iFindMinMax[{f, cons}, vars, opts]}, res /; ListQ[res]];
4 iFindMinMax[{ff_Max, cons_}, vars_, opts___?OptionQ] :=
5 Module[{z, res, f = List @@ ff},
6 res = FindMinimum[{z, (And @@ cons) && (And @@ Thread[z >= f])},
7 Append[Flatten[{vars}, 1], z], opts];
8 If[ListQ[res], {z /. res[[2]], Thread[vars -> (vars /. res[[2])]}]];
9 FindMinMax[{Max[{1/30 x + 1/15 y, 3/10 x + 1/5 y, 2 x + 1/2 y, x + y}], {x >=
10 0, y >= 0, 2 x + 2 y <= 10, -2 x - 2 y <= 0, y >= 1.5 x + 4}}, {x, y}]

```

And the output is

```
{4., {x -> 0., y -> 4.}}
```

Which is the same.

### 0.3 Problem 3

Barmish

### ECE 719 – Homework Diet

The table on the page to follow comes from a bestseller of the sixties, “Let’s Eat to Keep Fit,” by Adelle Davis. Let  $u_i$ , per labelling in the table, be the number of measures (quarts, cups cubes, etc.) of food  $i$  to be consumed. Now formulate a standard form linear programming problem to minimize the total cost of daily feeding while satisfying the following conditions:

- (i) Minimum and maximum daily allowances should be met; see table.
- (ii) Daily potassium intake should be within 15% of sodium intake.
- (ii) Daily calcium intake should be at least 75% of daily phosphorous intake.
- (a) Solve the LP above, indicating both the optimal solution and optimal cost.
- (b) Davis recommends that greater amounts of fat in the diet should be accompanied by greater vitamin B intake. To incorporate such a constraint let the amount of  $B_1$  and  $B_2$  be  $\gamma$ ( the amount of fat). Now, with this added constraint, re-run your LP for various values of  $\gamma$ .

Summarize all your work with appropriate commentary.

Food	Measure	Weight (g)	Price (\$Measure)	Minerals (mg)							Vitamins					
				Calories	Protein (g)	Fat (g)	Iron	Calcium	Phosphorus	Potassium	Sodium	VA (units)	VB <sub>1</sub> (mg)	VB <sub>2</sub> (mg)	VC (mg)	
<b>Dairy</b>																
1	Whole milk.....	1 qt.	976	0.40	660	32	40	0.4	1140	930	210	75	1,560	0.32	1.7	6
2	Ice cream.....	1 cup	188	0.35	300	6	18	0.1	175	150	170	140	740	0	0.3	0
3	Eggs (scrambled or fried) ..	2	128	0.15	220	13	16	2.2	60	222	140	338	1,200	0	0.4	0
4	Cheese (cheddar, American).....	1 in. cube	17	0.05	70	4	6	0.1	133	128	30	180	230	0	0.1	0
<b>Meat</b>																
5	Lean ground beef.....	3 oz.	85	0.25	185	24	10	3.0	10	158	340	110	20	0	0	0
6	Broiled chicken.....	3 oz.	85	0.12	185	23	9	1.4	10	250	350	50	260	0	0.1	0
7	Baked flounder.....	3.5 oz.	100	0.25	200	30	8	1.4	22	344	585	235	0	0	0	0
<b>Vegetable</b>																
8	French fried potatoes.....	10 pieces	60	0.07	155	1	7	0.7	9	6	510	6	0	0	0	8
<b>Fruit</b>																
9	Frozen o.j.....	6 oz. can	210	0.30	330	2	0	0.8	69	115	1,315	4	1,490	0.60	0.1	330
<b>Grain</b>																
10	Converted rice.....	1 cup uncooked	187	0.25	677	14	0	1.6	53	244	300	6	0	0.30	0	0
Min. daily allowance*.....					2,400	70	—	10	800	—	—	—	5,000	1.00	1.6	70
Max. daily allowance*.....					2,800	—	—	—	—	—	—	—	40,000	—	—	—

\* For healthy men, 35-55 years old, 5' 9" height, and 154 lb. weight.

Figure 8: problem 3 description

### 0.3.1 Part(a)

The variables are given in this table

variable	description	cost per unit	calories	potassium (mg)	sodium (mg)	calcium (mg)	phosphorus (mg)
$u_1$	whole milk	0.4	660	210	75	1140	930
$u_2$	ice cream	0.35	300	170	140	175	150
$u_3$	Eggs	0.15	220	140	338	60	222
$u_4$	Cheese	0.05	70	30	180	133	128
$u_5$	Beef	0.25	185	340	110	10	158
$u_6$	Broiled chicken	0.12	185	350	50	10	250
$u_7$	Baked vegetable	0.25	200	585	235	22	344
$u_8$	French fried	0.07	155	510	6	9	6
$u_9$	Frozen Grain	0.30	330	1315	4	69	115
$u_{10}$	Rice	0.25	677	300	6	53	244

Since the goal is to minimize the total cost of daily feeding, then

$$J(u) = 0.4u_1 + 0.35u_2 + 0.15u_3 + 0.05u_4 + 0.25u_5 + 0.12u_6 + 0.25u_7 + 0.07u_8 + 0.30u_9 + 0.25u_{10}$$

To be able to express the constraints, we need to first convert the minerals to same units used in  $J(u)$ . For example, one  $u_1$  is one qt, which is 976 grams. The mineral calcium is 1.14 gram, therefore in units of  $u_1$  it becomes  $\frac{1.14}{976}u_1$ , the same for all other minerals. Hence the above table becomes (where now each mineral is fraction of unit)

variable	description	cost per unit	calories	potassium)	sodium (mg)	calcium (mg)	phosphorus (mg)
$u_1$	whole milk	0.4	660	$\frac{0.210}{976}$	$\frac{0.075}{976}$	$\frac{1.140}{976}$	$\frac{0.930}{976}$
$u_2$	ice cream	0.35	300	$\frac{0.17}{188}$	$\frac{0.14}{188}$	$\frac{0.175}{188}$	$\frac{0.15}{188}$
$u_3$	Eggs	0.15	220	$\frac{0.14}{128}$	$\frac{0.338}{128}$	$\frac{0.06}{128}$	$\frac{0.222}{128}$
$u_4$	Cheese	0.05	70	$\frac{0.03}{17}$	$\frac{0.18}{17}$	$\frac{0.133}{17}$	$\frac{0.128}{17}$
$u_5$	Beef	0.25	185	$\frac{0.34}{85}$	$\frac{0.11}{85}$	$\frac{0.01}{85}$	$\frac{0.158}{85}$
$u_6$	Broiled chicken	0.12	185	$\frac{0.35}{85}$	$\frac{0.05}{85}$	$\frac{0.01}{85}$	$\frac{0.25}{85}$
$u_7$	Baked vegetable	0.25	200	$\frac{0.585}{100}$	$\frac{0.235}{100}$	$\frac{0.022}{100}$	$\frac{0.344}{100}$
$u_8$	French fried	0.07	155	$\frac{0.51}{60}$	$\frac{0.006}{60}$	$\frac{0.009}{60}$	$\frac{0.006}{60}$
$u_9$	Frozen Grain	0.30	330	$\frac{1.315}{210}$	$\frac{0.004}{210}$	$\frac{0.069}{210}$	$\frac{0.115}{210}$
$u_{10}$	Rice	0.25	677	$\frac{0.3}{187}$	$\frac{0.006}{187}$	$\frac{0.053}{187}$	$\frac{0.244}{187}$

Now we formulate each constraint. Let  $A$  be the total daily calories given by

$$A = 660u_1 + 300u_2 + 220u_3 + 70u_4 + 185u_5 + 185u_6 + 200u_7 + 155u_8 + 330u_9 + 677u_{10}$$

Hence constraint (i) in the problem becomes

$$A \geq 2400$$

$$A \leq 2800$$

For formulating constraint (ii), let  $B$  be the daily potassium and  $C$  be the daily sodium

$$B = \frac{0.210}{976}u_1 + \frac{0.17}{188}u_2 + \frac{0.14}{128}u_3 + \frac{0.03}{17}u_4 + \frac{0.34}{85}u_5 + \frac{0.35}{85}u_6 + \frac{0.585}{100}u_7 + \frac{0.51}{60}u_8 + \frac{1.315}{210}u_9 + \frac{0.3}{187}u_{10}$$

$$C = \frac{0.075}{976}u_1 + \frac{0.14}{188}u_2 + \frac{0.338}{128}u_3 + \frac{0.18}{17}u_4 + \frac{0.11}{85}u_5 + \frac{0.05}{85}u_6 + \frac{0.235}{100}u_7 + \frac{0.006}{60}u_8 + \frac{0.004}{210}u_9 + \frac{0.006}{187}u_{10}$$

Hence constraint (ii) in the problem becomes

$$B \leq 1.15C$$

$$B \geq 0.85C$$

For formulating constraint (iii), let  $D$  be the daily calcium and  $E$  be the daily phosphorus

$$D = \frac{1.140}{976}u_1 + \frac{0.175}{188}u_2 + \frac{0.06}{128}u_3 + \frac{0.133}{17}u_4 + \frac{0.01}{85}u_5 + \frac{0.01}{85}u_6 + \frac{0.022}{100}u_7 + \frac{0.009}{60}u_8 + \frac{0.069}{210}u_9 + \frac{0.053}{187}u_{10}$$

$$E = \frac{0.930}{976}u_1 + \frac{0.15}{188}u_2 + \frac{0.222}{128}u_3 + \frac{0.128}{17}u_4 + \frac{0.158}{85}u_5 + \frac{0.25}{85}u_6 + \frac{0.344}{100}u_7 + \frac{0.006}{60}u_8 + \frac{0.115}{210}u_9 + \frac{0.244}{187}u_{10}$$

Hence constraint (iii) in the problem becomes

$$D \geq 0.75E$$

Therefore the LP problem is

$$\begin{aligned} \min \quad & J(u) \\ \text{s.t.} \quad & A \geq 2400 \\ & A \leq 2800 \\ & B \leq 1.15C \\ & B \geq 0.85C \\ & D \geq 0.75E \end{aligned}$$

Converting to standard form and using  $x_i$  instead of  $u_i$  the above becomes

$$\begin{aligned} \min \quad & J(x) = 0.4x_1 + 0.35x_2 + 0.15x_3 + 0.05x_4 + 0.25x_5 + 0.12x_6 \\ & + 0.25x_7 + 0.07x_8 + 0.30x_9 + 0.25x_{10} \\ \text{subject to} \quad & A - \tilde{x}_{11} = 2400 \\ & A + \tilde{x}_{12} = 2800 \\ & B + \tilde{x}_{13} - 1.15C = 0 \\ & B - \tilde{x}_{14} - 0.85C = 0 \\ & D - \tilde{x}_{15} - 0.75E = 0 \end{aligned}$$

In full form, the above is

$$\begin{aligned}
\min \quad & J(x) = 0.4x_1 + 0.35x_2 + 0.15x_3 + 0.05x_4 + 0.25x_5 + 0.12x_6 + 0.25x_7 \\
\text{s.t.} \quad & 660x_1 + 300x_2 + 220x_3 + 70x_4 + 185x_5 + 185x_6 + 200x_7 + 155x_8 + 300x_9 \\
& 660x_1 + 300x_2 + 220x_3 + 70x_4 + 185x_5 + 185x_6 + 200x_7 + 155x_8 + 300x_9 \\
& \frac{0.210}{976}x_1 + \frac{0.17}{188}x_2 + \frac{0.14}{128}x_3 + \frac{0.03}{17}x_4 + \frac{0.34}{85}x_5 + \frac{0.35}{85}x_6 + \frac{0.585}{100}x_7 + \frac{0.51}{60}x_8 + \frac{1.315}{210}x_9 + \frac{0.3}{187}x_{10} + \tilde{x}_{13} - 1.15 \left( \frac{0.075}{976} \right) \\
& \frac{0.210}{976}x_1 + \frac{0.17}{188}x_2 + \frac{0.14}{128}x_3 + \frac{0.03}{17}x_4 + \frac{0.34}{85}x_5 + \frac{0.35}{85}x_6 + \frac{0.585}{100}x_7 + \frac{0.51}{60}x_8 + \frac{1.315}{210}x_9 + \frac{0.3}{187}x_{10} - \tilde{x}_{14} - 0.85 \left( \frac{0.075}{976} \right) \\
& \frac{1.140}{976}x_1 + \frac{0.175}{188}x_2 + \frac{0.06}{128}x_3 + \frac{0.133}{17}x_4 + \frac{0.01}{85}x_5 + \frac{0.01}{85}x_6 + \frac{0.022}{100}x_7 + \frac{0.009}{60}x_8 + \frac{0.069}{210}x_9 + \frac{0.053}{187}x_{10} - \tilde{x}_{15} - 0.75 \left( \frac{0.93}{976} \right)
\end{aligned}$$

Simplifying gives

$$\begin{aligned}
\min \quad & J(x) = 0.4x_1 + 0.35x_2 + 0.15x_3 + 0.05x_4 + 0.25x_5 + 0.12x_6 + 0.25x_7 \\
\text{s.t.} \quad & 660x_1 + 300x_2 + 220x_3 + 70x_4 + 185x_5 + 185x_6 + 200x_7 + 155x_8 + 300x_9 \\
& 660x_1 + 300x_2 + 220x_3 + 70x_4 + 185x_5 + 185x_6 + 200x_7 + 155x_8 + 300x_9 \\
& 1.268 \times 10^{-4}x_1 + 4.787 \times 10^{-5}x_2 - 1.943 \times 10^{-3}x_3 - 1.041 \times 10^{-2}x_4 + 2.512 \times 10^{-3}x_5 + 3.441 \times 10^{-3}x_6 + 3.117 \times 10^{-3}x_7 \\
& 1.499 \times 10^{-4}x_1 + 2.713 \times 10^{-4}x_2 - 1.151 \times 10^{-3}x_3 - 7.235 \times 10^{-3}x_4 + 0.003x_5 + 3.618 \times 10^{-3}x_6 + 3.853 \times 10^{-3}x_7 \\
& 4.534 \times 10^{-4}x_1 + 3.325 \times 10^{-4}x_2 - 8.32 \times 10^{-4}x_3 + 2.176 \times 10^{-3}x_4 - 1.277 \times 10^{-3}x_5 - 2.088 \times 10^{-3}x_6 - 0.001x_7
\end{aligned}$$

Therefore  $c^T x$  becomes

$$\min c^T x = \begin{pmatrix} 0.4 & 0.35 & 0.15 & 0.05 & 0.25 & 0.12 & 0.25 & 0.07 & 0.3 & 0.25 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \tilde{x}_{11} \\ \tilde{x}_{12} \\ \tilde{x}_{13} \\ \tilde{x}_{14} \\ \tilde{x}_{15} \end{pmatrix}$$

subject to  $Ax = b$

$$\begin{pmatrix}
 600 & 300 & 220 & 70 & 185 & 185 & 200 & 55 & 330 & 677 & -1 & 0 & 0 & 0 & 0 \\
 600 & 300 & 220 & 70 & 185 & 185 & 200 & 55 & 330 & 677 & 0 & 1 & 0 & 0 & 0 \\
 1.267 \times 10^{-4} & 4.787 \times 10^{-5} & -1.943 \times 10^{-3} & -1.041 \times 10^{-2} & 2.512 \times 10^{-3} & 3.441 \times 10^{-3} & 3.148 \times 10^{-3} & 8.385 \times 10^{-3} & 0.006 & 1.567 \times 10^{-3} & 0 & 0 & 1 & 0 & 0 \\
 1.498 \times 10^{-4} & 2.712 \times 10^{-4} & -1.150 \times 10^{-3} & -7.235 \times 10^{-3} & 0.003 & 3.617 \times 10^{-3} & 3.852 \times 10^{-3} & 8.415 \times 10^{-3} & 6.245 \times 10^{-3} & 1.577 \times 10^{-3} & 0 & 0 & 0 & -1 & 0 \\
 4.534 \times 10^{-4} & 3.324 \times 10^{-4} & -8.320 \times 10^{-4} & 2.176 \times 10^{-3} & -1.276 \times 10^{-3} & -2.088 \times 10^{-3} & -0.002 & 7.5 \times 10^{-5} & -8.214 \times 10^{-5} & -6.952 \times 10^{-4} & 0 & 0 & 0 & 0 & -1
 \end{pmatrix}
 \begin{pmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6 \\
 x_7 \\
 x_8 \\
 x_9 \\
 x_{10} \\
 \tilde{x}_{11} \\
 \tilde{x}_{12} \\
 \tilde{x}_{13} \\
 \tilde{x}_{14} \\
 \tilde{x}_{15}
 \end{pmatrix}
 =
 \begin{pmatrix}
 2400 \\
 2800 \\
 0 \\
 0 \\
 0
 \end{pmatrix}$$

The above is now solved using Matlab linprog. The following is the result, followed by the Matlab code used.

$$x_{optimal} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \tilde{x}_{11} \\ \tilde{x}_{12} \\ \tilde{x}_{13} \\ \tilde{x}_{14} \\ \tilde{x}_{15} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1.0792 \\ 0 \\ 0 \\ 0 \\ 0.2889 \\ 0 \\ 3.41 \\ 0 \\ 400 \\ 0.0035 \\ 0 \\ 0 \end{pmatrix}$$

In terms of raw  $u_i$  variables, the above says to buy <sup>1</sup>

1.0792 cube of cheese, and 2.889 pieces of french fried and 3.41 rice cups

### Verification:

Calories from the above is

$$(1.0792)(70) + (0.2889)(155) + (3.41)(677) = 2428.9$$

<sup>1</sup>This seems to meet the requirements, but it does not look like a healthy diet actually?

Which is within daily allowance OK. Potassium from the above is

$$(1.0792) \left( \frac{0.03}{17} \right) + (0.2889) \left( \frac{0.51}{60} \right) + (3.41) \left( \frac{0.3}{187} \right) = 9.831 \text{ mg}$$

While sodium from the above is

$$(1.0792) \left( \frac{0.18}{17} \right) + (0.2889) \left( \frac{0.006}{60} \right) + (3.41) \left( \frac{0.006}{187} \right) = 10.1565 \text{ mg}$$

Hence Potassium is within 15% of sodium OK. Daily calcium from above is

$$(1.0792) \left( \frac{0.133}{17} \right) + (0.2889) \left( \frac{0.009}{60} \right) + (3.41) \left( \frac{0.053}{187} \right) = 9.453 \text{ mg}$$

While daily phosphorus is

$$(1.0792) \left( \frac{0.128}{17} \right) + (0.2889) \left( \frac{0.006}{60} \right) + (3.41) \left( \frac{0.244}{187} \right) = 12.6 \text{ mg}$$

Hence Daily calcium is at least 75% of daily phosphorus OK. All verified OK. The corresponding optimal daily cost is

$$J^* = 0.9267 \text{ dollars}$$

```

1 function nma_HW6_problem_3
2 %Solves third problem, HW 6, ECE 719
3 %Nasser M. Abbasi
4
5 c = [0.4 0.35 0.15 0.05 0.25 0.12 0.25 0.07 0.3 0.25 0 0 0 0 0];
6 A = [600 300 220 70 185 185 200 55 330 677 -1 0 0 0 0;
7      600 300 220 70 185 185 200 55 330 677 0 1 0 0 0;
8      1.2679*10^(-4), 4.7872*10^(-5), -1.9430*10^(-3), ...
9      -1.0412*10^(-2), ...
10     2.5118*10^(-3), 3.4412*10^(-3), 3.1475*10^(-3), 8.385*10^(-3), ...
11     0.00624, 1.5674*10^(-3), 0, 0, 1, 0, 0;
12     1.4985*10^(-4), 2.7128*10^(-4), -1.1508*10^(-3), ...
13     -7.2353*10^(-3), ...
14     0.0029, 3.6176*10^(-3), 3.8525*10^(-3), 8.415*10^(-3), ...
15     6.2457*10^(-3), ...
16     1.577*10^(-3), 0, 0, 0, -1, 0;
17     4.5338*10^(-4), 3.3245*10^(-4), -8.3203*10^(-4), ...
18     2.1765*10^(-3), ...
19     -1.2765*10^(-3), -2.0882*10^(-3), -0.00236, 7.5*10^(-5), ...
20     -8.2143*10^(-5), ...
21     -6.9519*10^(-4), 0, 0, 0, 0, -1];
22 b=[2400 2800 0 0 0];
23 options = optimset('LargeScale','off','Simplex','on');
24 [X,FVAL,EXITFLAG,OUTPUT]=linprog(c,[],[],A,b,zeros(size(c)),...
25                                 [],[],options)
26 end

```

The output from the above is

```

Optimization terminated.
X =
0
0
0
1.0792
0
0
0
0
0.2889
0
3.4100
0
400.0000
0.0035
0
0
FVAL =
0.9267
EXITFLAG =
1
OUTPUT =
iterations: 0
algorithm: 'simplex'
cgiterations: []
message: 'Optimization terminated.'
constrviolation: 4.3368e-19
firstorderopt: 5.5511e-17

```

### 0.3.2 part(b)

Now we add a new constraint, which is  $B_1 + B_2 = \gamma(fat)$ . In terms of  $u_i$ , this becomes

$$B_1 = \frac{0.00032}{976}u_1 + \frac{0.0006}{210}u_9 + \frac{0.0003}{187}u_{10}$$

And

$$B_2 = \frac{0.0017}{976}u_1 + \frac{0.0003}{188}u_2 + \frac{0.0004}{128}u_3 + \frac{0.0001}{17}u_4 + \frac{0.0001}{85}u_6 + \frac{0.0001}{210}u_9$$

And

$$fat = \frac{40}{976}u_1 + \frac{18}{188}u_2 + \frac{16}{128}u_3 + \frac{6}{17}u_4 + \frac{10}{85}u_5 + \frac{9}{85}u_6 + \frac{8}{100}u_7 + \frac{7}{60}u_8$$

Hence the new constraint is

$$(B_1 + B_2) - \gamma(fat) = 0$$



Converting to standard LP form, the above becomes

$$\left(2.06976 \times 10^{-6} - \frac{5}{122}\gamma\right)x_1 + \left(1.59574 \times 10^{-6} - \frac{9}{94}\gamma\right)x_2 + \left(3.125 \times 10^{-6} - \frac{1}{8}\gamma\right)x_3 + \left(5.88235 \times 10^{-6} - \frac{6}{17}\gamma\right)x_4 - \frac{2}{17}\gamma x_5 + \left(1.17647 \times 10^{-6} - \frac{9}{85}\gamma\right)x_6 - \frac{2}{25}\gamma x_7 - \frac{7}{60}\gamma x_8 + 3.33333 \times 10^{-6}x_9 + 1.604 \times 10^{-6}\gamma x_{10} = 0$$

Now the new  $Ax = b$  becomes ( $c^T x$  do not change from part(a) and remain the same).

$$\begin{pmatrix} 600 & 300 & 220 & 70 & 185 & 185 & 200 & 55 & 330 & 677 & -1 & 0 & 0 & 0 & 0 \\ 600 & 300 & 220 & 70 & 185 & 185 & 200 & 55 & 330 & 677 & 0 & 1 & 0 & 0 & 0 \\ 1.268 \times 10^{-4} & 4.787 \times 10^{-5} & -1.943 \times 10^{-3} & -1.041 \times 10^{-2} & 2.512 \times 10^{-3} & 3.441 \times 10^{-3} & 3.148 \times 10^{-3} & 8.385 \times 10^{-3} & 0.006 & 1.567 \times 10^{-3} & 0 & 0 & 1 & 0 & 0 \\ 1.499 \times 10^{-4} & 2.713 \times 10^{-4} & -1.151 \times 10^{-3} & -7.235 \times 10^{-3} & 0.003 & 3.618 \times 10^{-3} & 3.853 \times 10^{-3} & 8.415 \times 10^{-3} & 6.2467 \times 10^{-3} & 1.577 \times 10^{-3} & 0 & 0 & 0 & -1 & 0 \\ 4.534 \times 10^{-4} & 3.325 \times 10^{-4} & -8.32 \times 10^{-4} & 2.177 \times 10^{-3} & -1.277 \times 10^{-3} & -2.088 \times 10^{-3} & -0.003 & 7.5 \times 10^{-5} & -8.214 \times 10^{-5} & -6.982 \times 10^{-4} & 0 & 0 & 0 & 0 & -1 \\ 2.06976 \times 10^{-6} - 0.03279\gamma & (1.59574 \times 10^{-6} - 0.031915\gamma) & (3.125 \times 10^{-6} - 0.1016\gamma) & (5.88235 \times 10^{-6} - 0.2353\gamma) & -0.28235\gamma & (1.17647 \times 10^{-6} - 0.27058\gamma) & -0.3\gamma & -0.01667\gamma & (3.33333 \times 10^{-6} - 0.00953\gamma) & -0.0749\gamma & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \tilde{x}_{11} \\ \tilde{x}_{12} \\ \tilde{x}_{13} \\ \tilde{x}_{14} \\ \tilde{x}_{15} \end{pmatrix} = \begin{pmatrix} 2400 \\ 2800 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The above was solve in Matlab for different values of  $\gamma$ .For  $\gamma = 0.00001$ , the optimal  $x$  was

$$x_{optimal} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \tilde{x}_{11} \\ \tilde{x}_{12} \\ \tilde{x}_{13} \\ \tilde{x}_{14} \\ \tilde{x}_{15} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 4.2150 \\ 6.8174 \\ 0 \\ 0 \\ 3.0043 \\ 0 \\ 1.0022 \\ 0 \\ 400 \\ 0 \\ 0.0161 \\ 0 \end{pmatrix}$$

With optimal  $J^* = 2.376$  dollars. For  $\gamma = 0.00002$ , the optimal  $x$  was

$$x_{optimal} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \tilde{x}_{11} \\ \tilde{x}_{12} \\ \tilde{x}_{13} \\ \tilde{x}_{14} \\ \tilde{x}_{15} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2.0750 \\ 0 \\ 0 \\ 0 \\ 1.1779 \\ 0 \\ 3.2348 \\ 0 \\ 400 \\ 0.0067 \\ 0 \\ 0.0024 \end{pmatrix}$$

With optimal  $J^* = 0.9949$  dollars. For  $\gamma = 0.00003$ , the optimal  $x$  was

$$x_{optimal} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \tilde{x}_{11} \\ \tilde{x}_{12} \\ \tilde{x}_{13} \\ \tilde{x}_{14} \\ \tilde{x}_{15} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1.0712 \\ 0 \\ 0 \\ 0 \\ 0.2078 \\ 0.1119 \\ 3.3629 \\ 0 \\ 400 \\ 0.0034 \\ 0 \\ 0 \end{pmatrix}$$

With optimal  $J^* = 0.9424$  dollars. For  $\gamma = 0.00004$ , the optimal  $x$  was

$$x_{optimal} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \tilde{x}_{11} \\ \tilde{x}_{12} \\ \tilde{x}_{13} \\ \tilde{x}_{14} \\ \tilde{x}_{15} \end{pmatrix} = \begin{pmatrix} 0.3431 \\ 0 \\ 0 \\ 0.8519 \\ 0 \\ 0 \\ 0 \\ 0.7094 \\ 2.8071 \\ 0 \\ 0 \\ 400 \\ 0 \\ 0.0027 \\ 0 \end{pmatrix}$$

With optimal  $J^* = 1.0944$  dollars. When going more than  $\gamma = 0.00004$ , say  $\gamma = 0.00005$ , Matlab was not able to obtain solution using simplex method. The message was "Exiting: The constraints are overly stringent; no feasible starting point found"

Exiting: The constraints are overly stringent;  
no feasible starting point found.

```
X =
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
-2400
2800
0
0
0
0
FVAL =
0
```

