190031187 NERELLA VENKATA RADHAKRISHNA

MP-1 TUTORIAL-3

PRELAB

1. Differentiate Simplex and two-phase simplex method.

MP-1 Tutorial - 3

190031187
Radhakrushna

Prelab

1: Two phase method differs from simplex method that first it accomplishes an auxillary problem that has to minimize the sum of artificial variables. once the first problem is solved, we start with second phase that consists in making a normal simplex.

when using simplex method for greater than or equal to constraint the slack variable has a negative coefficient and equality constraints do not have slack variables. If either of constraint is part of the model there is no convinient IBFS and hence two phase method is used.

phase I : Minimize the sum of the artificial variables

phase-II : use the bfs obtained after the completion of the phase I, as a starting bfs for phase II for optimal solution.

2. Minimum Z=x1+x2
   Subject to:
   2x1 + x2 >=4
   X1 + 7x2 >=7
   And x1, x2>=0
   Solve using Two-phase method

2. Converting minimization to maximization

maximize $z = -x_1 + (-x_2)$

Subject to : $2x_1 + x_2 \geq 4$

$x_1 + 7x_2 \geq 7$

$x_1, x_2 \geq 0$

converting inequalities to equalities

$2x_1 + x_2 - s_1 + A_1 = 4$

$x_1 + 7x_2 - s_2 + A_2 = 7$

phase - I

Maximize $0x_1 + 0x_2 + 0s_1 + 0s_2 - A_1 - A_2$

Subject to

$2x_1 + x_2 - s_1 + A_1 = 4$

$x_1 + 7x_2 - s_2 + A_2 = 7$

Initial Table

C_{Bi}	C_j	0	0	0	0	-1	-1	
	B.V	x_1	x_2	s_1	s_2	A_1	A_2	Sol
-1	A_1	2	1	-1	0	1	0	4
-1	A_2	1	7	0	-1	0	1	7
	Z_j	-3	-8	1	1	-1	-1	
	$C_j - Z_j$	3	8	-1	-1	0	0	

2. Entering variable = x_2
leaving variable = A_2
key element = 7

Iteration - 1

C_{Bi}	C_j	0	0	0	0	-1	-1	
	B.V	x_1	x_2	S_1	S_2	A_1	A_2	sol
-1	A_1	13/7	0	-1	1/7	1	-1/7	3
0	x_2	1/7	1	0	-1/7	0	1/7	1
	Z_j	-13/7	0	1	-1/7	-1	1/7	
	$C_j - Z_j$	13/7	0	-1	1/7	0	-8/7	

Entering variable : x_1
leaving variable : A_1
keyelement : 13/7

Iteration - II

C_{Bi}	C_j	0	0	0	0	-1	-1	
	B.V	x_1	x_2	S_1	S_2	A_1	A_2	sol
0	x_1	1	0	-7/13	1/13	7/13	-1/13	21/13
0	x_2	0	1	1/13	-14/91	-1/13	-12/91	10/13
	Z_j	0	0	0	0	0	0	0
	$C_j - Z_j$	0	0	0	0	-1	-1	

phase-1 terminates because both the artificial variables have been removed from the basis

phase-2:

C_B	c_j	-1	-1	0	0	
	BV	x_1	x_2	S_1	S_2	sol
-1	x_1	1	0	-7/13	1/13	21/13
-1	x_2	0	1	1/13	-14/91	10/13
	z_j	-1	-1	6/13	1/13	-31/13
	c_j-z_j	0	0	-6/13	-1/13	

$x_1 = 21/13$

$x_2 = 10/3$

(maximization) $Z = -31/13$

Since the problem is of minimization

$$\text{minimize} \sum_{j=1}^{n} c_j x_j = \text{maximize} \sum_{j=1}^{n} (-c_j) x_j$$

Hence z (minimize) $= 31/13$

$x_1 = 21/13$

$x_2 = 10/13$

INLAB

1.Minimize: z = x1 + x2 + x3 + x4 + x5

Subject to:

3x1 + 2x2 + x3 = 1
5x1 + x2 + x3 + x4 = 3
2x1 + 5x2 + x3 + x5 = 4

Solve using two-phase simplex method in python.

```
def printTableu(tableu):
 print '----------------------'
 for row in tableu:
  print row
 print '----------------------'
 return

def pivotOn(tableu, row, col):
 j = 0
 pivot = tableu[row][col]
 for x in tableu[row]:
  tableu[row][j] = tableu[row][j] / pivot
  j += 1
 i = 0
 for xi in tableu:
  if i != row:
   ratio = xi[col]
   j = 0
   for xij in xi:
    xij -= ratio * tableu[row][j]
    tableu[i][j] = xij
    j += 1
  i += 1
 return tableu

# assuming tablue in standard form with basis formed in last m columns
def phase_1_simplex(tableu):

 THETA_INFINITE = -1
 opt   = False
 unbounded  = False
 n = len(tableu[0])
 m = len(tableu) - 2

 while ((not opt) and (not unbounded)):
  min = 0.0
  pivotCol = j = 1
  while(j < (n-m)):
   cj = tableu[1][j]
   if (cj < min):
    min = cj
    pivotCol = j
   j += 1
```

```
  if min == 0.0:
   opt = True
   continue
  pivotRow = i = 0
  minTheta = THETA_INFINITE
  for xi in tableu:
   if (i > 1):
    xij = xi[pivotCol]
    if xij > 0:
     theta = (xi[0] / xij)
     if (theta < minTheta) or (minTheta == THETA_INFINITE):
      minTheta = theta
      pivotRow = i
   i += 1
  if minTheta == THETA_INFINITE:
   unbounded = True
   continue
  tableu = pivotOn(tableu, pivotRow, pivotCol)
 return tableu

def simplex(tableu):
 THETA_INFINITE = -1
 opt   = False
 unbounded  = False
 n = len(tableu[0])
 m = len(tableu) - 1

 while ((not opt) and (not unbounded)):
  min = 0.0
  pivotCol = j = 0
  while(j < (n-m)):
   cj = tableu[0][j]
   if (cj < min) and (j > 0):
    min = cj
    pivotCol = j
   j += 1
  if min == 0.0:
   opt = True
   continue
  pivotRow = i = 0
  minTheta = THETA_INFINITE
  for xi in tableu:
   if (i > 0):
    xij = xi[pivotCol]
    if xij > 0:
     theta = (xi[0] / xij)
     if (theta < minTheta) or (minTheta == THETA_INFINITE):
      minTheta = theta
      pivotRow = i
   i += 1
  if minTheta == THETA_INFINITE:
```

```
   unbounded = True
   continue
  tableu = pivotOn(tableu, pivotRow, pivotCol)
 return tableu

def drive_out_artificial_basis(tableu):
 n = len(tableu[0])
 j = n - 1
 isbasis = True
 while(j > 0):
  found = False
  i = -1
  row = 0
  for xi in tableu:
   i += 1
   if (xi[j] == 1):
    if (found):
     isbasis = False
     continue
    elif (i > 1):
     row = i
     found = True
   elif (xi[0] != 0):
    isbasis = False
    continue
  if (isbasis and found):
   if (j >= n):
    tableu = pivotOn(tableu, row, j)
   else:
    return tableu
  j -= 1
 return tableu

def two_phase_simpelx(tableu):
 infeasible  = False
 tableu = phase_1_simplex(tableu)
 sigma = tableu[1][0]
 if (sigma > 0):
  infeasible  = True
  print 'infeasible'
 else:
  #sigma is equals to zero
  tableu = drive_out_artificial_basis(tableu)
  m = len(tableu) - 2
  n = len(tableu[0])
  n -= m
  tableu.pop(1)
  i = 0
  while (i < len(tableu)):
   tableu[i] = tableu[i][:n]
   i += 1
```

```
  tableu = simplex(tableu)
 return tableu

def getTableu(c, eqs, b):
 #assume b >= 0 so if there is any b[i] negative make sure to enter
 #it possitive by multiplying (-1 * eqs[i]) and (-1 * b[i]) for all i
 tableu = []
 m = len(eqs)
 n = len(c)
 c.insert(0, 0.0)
 artificial = []
 sigma = [0.0]
 i = 0
 while (i < n):
  sigma.append(0.0)
  i += 1
 i = 0
 while (i < m):
  artificial.append(0.0)
  sigma.append(1.0)
  i += 1
 c.extend(artificial)
 tableu.append(c)
 tableu.append(sigma)
 i = 0
 for eq in eqs:
  eq.insert(0, b[i])
  eq.extend(artificial)
  eq[n+1+i] = 1.0
  tableu.append(eq)
  i += 1
 i = 0
 for xi in tableu:
  if (i > 1):
   j = 0
   for xij in xi:
    tableu[1][j] -= xij
    j += 1
  i += 1
 return tableu

c = [ 1.0, 1.0, 1.0, 1.0, 1.0,]
eq1 = [ 3.0 ,  2.0 , 1.0 ,  0.0,  0.0]
eq2 = [ 5.0 ,  1.0 , 1.0 ,  1.0,  0.0]
eq3 = [ 2.0 ,  5.0 , 1.0 ,  0.0,  1.0]
b = [1.0 , 3.0 , 4.0]
eqs = []
eqs.append(eq1)
eqs.append(eq2)
eqs.append(eq3)
tableu = getTableu(c,eqs,b)
```

printTableu(tableu)
tableu = two_phase_simpelx(tableu)
printTableu(tableu)
print 'minimum cost is = {}'.format(-tableu[0][0])

Output :

```
c = [ 1.0, 1.0, 1.0, 1.0, 1.0,]
eq1 = [ 3.0 ,  2.0 , 1.0 ,  0.0,  0.0]
eq2 = [ 5.0 ,  1.0 , 1.0 ,  1.0,  0.0]
eq3 = [ 2.0 ,  5.0 , 1.0 ,  0.0,  1.0]

b = [1.0 , 3.0 , 4.0]

eqs = []
eqs.append(eq1)
eqs.append(eq2)
eqs.append(eq3)

tableu = getTableu(c,eqs,b)
printTableu(tableu)
tableu = two_phase_simpelx(tableu)

printTableu(tableu)
print 'minimum cost is = {}'.format( -tableu[0][0])
```

```
----------------------
[0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0]
[-8.0, -10.0, -8.0, -3.0, -1.0, -1.0, 0.0, 0.0, 0.0]
[1.0, 3.0, 2.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0]
[3.0, 5.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 0.0]
[4.0, 2.0, 5.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0]
----------------------
----------------------
[-4.5, 1.5000000000000004, 0.0, 1.5, 0.0, 0.0]
[0.5, 1.5, 1.0, 0.5, 0.0, 0.0]
[2.5, 3.4999999999999996, 0.0, 0.5, 1.0, 0.0]
[1.5, -5.5, 0.0, -1.5, 0.0, 1.0]
----------------------
minimum cost is = 4.5
```

2. Minimize $z = -3p_1 + p_2 - 2p_3$
subject to
$p_1 + 3p_2 + p_3 \le 5$
$2p_1 - p_2 + p_3 \ge 2$
$4p_1 + 3p_2 - 2p_3 = 5$
$x_1, x_2, x_3 \ge 0$

Inlab

2. let P_1 be x_1, P_2 be x_2, P_3 be x_3

converting minimization into maximization

maximize $z = 3x_1 - x_2 + 2x_3$

Subject to:

$x_1 + 3x_2 + x_3 \le 5$

$2x_1 - x_2 + x_3 \ge 2$

$4x_1 + 3x_2 - 2x_3 = 5$

converting inequalities to equalities

$x_1 + 3x_2 + x_3 + S_1 = 5$

$2x_1 - x_2 + x_3 - S_2 + A_1 = 2$

$4x_1 + 3x_2 - 2x_3 + A_2 = 5$

Initial Table :-

c_{Bi}	c_j	0	0	0	0	0	-1	-1	
	B.V	x_1	x_2	x_3	S_1	S_2	A_1	A_2	Sol
0	S_1	1	3	1	1	0	0	0	5
-1	A_1	2	-1	1	0	-1	1	0	2
-1	A_2	4	3	-2	0	0	0	1	5
	z_j	-6	-2	1	0	1	-1	-1	-4
	$c_j - z_j$	6	2	-1	0	-1	0	0	

Iteration -1

C_{B_i}	C_j	0	0	0	0	0	-1	-1	
	B.V	x_1	x_2	x_3	S_1	S_2	A_1	A_2	sol
0	S_1	0	$7/2$	$1/2$	1	$1/2$	$-1/2$	0	4
0	x_1	1	$-1/2$	$1/2$	0	$-1/2$	$1/2$	0	1
-1	A_2	0	5	-4	0	2	-2	1	1
	z_j	0	-5	4	0	-2	2	-1	
	$c_j - z_j$	0	5	-4	0	2	-3	0	

Key element : 5

Iteration - II

C_{B_i}	C_j	0	0	0	0	0	-1	-1	
	B.V	x_1	x_2	x_3	S_1	S_2	A_1	A_2	sol
0	S_1	0	0	$\frac{33}{10}$	1	$\frac{-9}{10}$	$\frac{9}{10}$	$\frac{-7}{10}$	$\frac{33}{10}$
0	x_1	1	0	$1/10$	0	$-3/10$	$3/10$	$1/10$	$11/10$
0	x_2	0	1	$-4/5$	0	$2/5$	$-2/5$	$1/5$	$1/5$
	z_j	0	0	0	0	0	0	0	0
	$c_j - z_j$	0	0	0	0	0	-1	-1	

phase -I terminates because both the artificial variables are eliminated

phase-2

C_{B_i}	C_j	3	-1	2	0	0	
	B.V	x_1	x_2	x_3	s_1	s_2	Sol
0	s_1	0	0	$\frac{33}{10}$	1	$\frac{-9}{10}$	$\frac{33}{10}$
3	x_1	1	0	$1/10$	0	$-3/10$	$11/10$
-1	x_2	0	1	$-4/5$	0	$2/5$	$1/5$
	z_j	3	-1	$11/10$	0	$-13/10$	
	$c_j - z_j$	0	0	$9/10$	0	$13/10$	

Iteration-1

C_{B_i}	C_j	3	-1	2	0	0	
	B.V	x_1	x_2	x_3	s_1	s_2	Sol
0	s_1	0	$9/4$	$3/2$	1	0	$15/4$
3	x_1	1	$3/4$	$-1/2$	0	0	$5/4$
0	s_2	0	$5/2$	-2	0	1	$1/2$
	z_j	3	$9/4$	$-3/2$	0	0	
	$c_j - z_j$	0	$-13/4$	$7/2$	0	0	

C_{B_i}	C_j	3	-1	2	0	0	
	B.V	x_1	x_2	x_3	S_1	S_2	Sol
2	x_3	0	$3/2$	1	$2/3$	0	$5/2$
3	x_1	1	$3/2$	0	$1/3$	0	$5/2$
0	S_2	0	$1/2$	0	$4/3$	1	$11/2$
	z_j	3	$15/2$	2	$7/3$	0	$25/2$
	$C_j - z_j$	0	$-17/2$	0	$-7/3$	0	

$\therefore$ $x_1 = 5/2$

$x_2 = 0$

$x_3 = 5/2$

Since the problem is of minimization

$z(\text{minimized}) = -25/2$

$x_1 = 5/2$

$x_2 = 0$

$x_3 = 5/2$

POSTLAB

1. Minimize Z =5 x1 + 2x2 +10 x3

Subject to :

X1 – x3 <=10

X2 + x3 >=10

And x1,x2,x3 >=0

Solve using two-phase simplex method.

post-lab

1. Minimize $z = 5x_1 + 2x_2 + 10x_3$

subject to:

$x_1 - x_3 \leq 10$

$x_2 + x_3 \geq 10$

$x_1, x_2, x_3 \geq 0$

Maximize $z = -5x_1 - 2x_2 - 10x_3$

subject to:

$x_1 - x_3 \leq 10$

$x_2 + x_3 \geq 10$

$x_1, x_2, x_3 \geq 0$

converting inequalities to equalities

$x_1 - x_3 + s_1 = 10$

$x_2 + x_3 - s_2 + A_1 = 10$

phase-1

Maximize $z = 0x_1 + 0x_2 + 0x_3 + 0s_1 + 0s_2 - A_1$

subject to:

$x_1 - x_3 + s_1 = 10$

$x_2 + x_3 - s_2 + A_1 = 10$

$x_1, x_2, x_3, s_1, s_2, A_1 \geq 0$

Intial Table

C_{Bi}	C_j	0	0	0	0	0	-1	
	B.V	x_1	x_2	x_3	S_1	S_2	A_1	Sol
0	S_1	1	0	-1	1	0	0	10
-1	A_1	0	1	1	0	-1	1	10
	Z_j	0	-1	-1	0	1	-1	-10
	$C_j - Z_j$	0	1	1	0	-1	0	

key element = 1

Iteration -1

C_{Bi}	C_j	0	0	0	0	0	-1	
	B.V	x_1	x_2	x_3	S_1	S_2	A_1	sol
0	S_1	1	0	-1	1	0	0	10
0	x_2	0	1	1	0	-1	1	10
	Z_j	0	0	0	0	0	0	0
	$C_j - Z_j$	0	0	0	0	0	-1	

phase - I terminates because all the artificial variables are elimin-ated phase - II

$$\text{Max } z = -5x_1 - 2x_1 - 10x_3 + 0s_1 + 0s_2$$

C_{Bi}	C_j	-5	-2	-10	0	0	
	BV	x_1	x_2	x_3	s_1	s_2	sol
0	s_1	1	0	-1	-1	0	10
-2	x_2	0	1	1	0	-1	10
	z_j	0	-2	-2	0	2	-20
	$c_j - z_j$	-5	0	-8	0	-2	

since the problem is of minimization.

$z = 20$
$x_1 = 0$
$x_2 = 10$
$x_3 = 0$

2. Maximize $z = 12a_1 + 15a_2 + 9a_3$
subject to
$8a_1 + 16a_2 + 12a_3 \le 250$
$4a_1 + 8a_2 + 10a_3 \ge 80$
$7a_1 + 9a_2 + 8a_3 = 105$
$a_1, a_2, a_3 \ge 0$

2. let a_1 be x_1, a_2 be x_2, a_3 be x_3

Max $z = 12x_1 + 15x_2 + 9x_3$
subject to

$8x_1 + 16x_2 + 12x_3 \le 250$
$4x_1 + 8x_2 + 10x_3 \ge 80$
$7x_1 + 9x_2 + 8x_3 = 105$

converting inequalities to equalities

$8x_1 + 16x_2 + 12x_3 + S_1 = 250$
$4x_1 + 8x_2 + 10x_3 - S_2 + A_1 = 80$
$7x_1 + 9x_2 + 8x_3 + A_2 = 105$

phase-I

C_{Bi}	C_j	0	0	0	0	0	-1	-1	
	B.V	x_1	x_2	x_3	S_1	S_2	A_1	A_2	sol
0	S_1	8	16	12	1	0	0	0	250
-1	A_1	4	8	10	0	-1	1	0	80
-1	A_2	7	9	8	0	0	0	1	105
	z_j	-11	-17	-18	0	1	-1	-1	
	$C_j - z_j$	11	17	18	0	-1	0	0	

Iteration-1

C_{B_i}	C_j	0	0	0	0	0	-1	-1	
	B·V	x_1	x_2	x_3	S_1	S_2	A_1	A_2	sol
0	S_1	16/5	32/5	0	1	6/5	-6/5	0	154
0	x_3	2/5	4/5	1	0	-1/10	1/10	0	8
-1	A_2	19/5	13/5	0	0	4/5	-4/5	1	41
	Z_j	-19/5	-13/5	0	0	-4/5	4/5	-1	
	C_j-Z_j	19/5	13/5	0	0	4/5	-9/5	0	

Iteration-2

C_{B_i}	C_j	0	0	0	0	0	-1	-1	
	B·V	x_1	x_2	x_3	S_1	S_2	A_1	A_2	Sol
0	S_1	0	80/19	0	1	10/19	$\frac{-10}{19}$	$-\frac{16}{19}$	$\frac{2270}{19}$
0	x_3	0	10/19	1	0	-7/38	7/38	$-\frac{2}{19}$	70/19
0	x_1	1	13/19	0	0	4/19	-4/19	5/19	$\frac{205}{19}$
	Z_j	0	0	0	0	0	0	0	
	C_j-Z_j	0	0	0	0	0	-1	-1	

phase-I terminates because all the artificial variables phase - II

$\text{Max } z = 12x_1 + 15x_2 + 9x_3 + 0s_1 + 0s_2$

C_{Bi}	C_j	12	15	9	0	0	
	B.V	x_1	x_2	x_3	s_1	s_2	sol
0	s_1	0	80/19	0	1	10/19	2270/19
9	x_3	0	60/19	1	0	-7/38	70/19
12	x_1	1	13/19	0	0	4/19	205/19
	z_j	12	246/19	9	0	33/38	
	$C_j - z_j$	0	39/19	0	0	-33/38	

C_{Bi}	C_j	12	15	9	0	0	
	B.V	x_1	x_2	x_3	s_1	s_2	sol
0	s_1	0	0	-8	1	2	90
15	x_2	0	1	19/10	0	-7/20	7
12	x_1	1	0	-13/10	0	9/20	6
	z_j	12	15	129/10	0	3/20	177
	$C_j - z_j$	0	0	-3/10	0	-3/20	

$\therefore$ The optimal solution is

$x_1 = 6$

$x_2 = 7$

$x_3 = 0$ Max $z = 177$