

# LINEAR PROGRAMMING

1.224J/ESD.204J  
TRANSPORTATION OPERATIONS,  
PLANNING AND CONTROL:  
CARRIER SYSTEMS

Professor Cynthia Barnhart  
Professor Nigel H.M. Wilson

*Fall 2003*

# Announcements

- Reader
- Problem set #1
- December 1 recitation & December 5 class
- OPL Studio examples

# LINEAR PROGRAMMING

Sources:

- Introduction to linear optimization (Bertsimas, Tsitsiklis)
- Nathaniel Grier's paper
- 1.224 previous material

# Outline

1. Modeling problems as linear programs
2. Solving linear programs

# Outline

1. Modeling problems as linear programs
  - What is a linear Program
  - Formulation
  - Set Notation Review
  - Example: Transit Ridership
  - Standard Form of an LP
  - Linearity
  - Examples

# What is a Linear Program (LP)?

## 1. Objective Function

- summarizes objective of the problem (MAX, MIN)

## 2. Constraints of problem:

- limitations placed on the problem; control allowable solutions
- Problem statement: ‘given....’, ‘must ensure...’, ‘subject to’
- Equations or inequalities

## 3. Decision Variables

- quantities, decisions to be determined
- multiple types (real numbers, non-negative, integer, binary)
- In an LP, the decision variables are real numbers
- Choice of decision variables will determine difficulty in formulating and solving the problem

# Set Notation Review

Set: collection of distinct objects

R: set of real numbers

Z: set of integers

$\emptyset$ : empty set

Superscript +: non-negative elements of a set

$\in$  : 'is an element of'

$\{ \}$  : 'the set containing' (members of the set are between brackets)

: or  $|$  : 'such that' example:  $\{x \in S : x \geq 0\}$

$\exists$  : 'there exists'

$\forall$  : 'for all'

# Example: Transit Ridership

- A transit agency is performing a review of the service it provides. It has decided to measure its overall effectiveness in terms of the total number of riders it serves. The agency operates a number of modes of transport. The table shows the average number of riders generated by each trip (by mode) and the cost of each trip (by mode)

Mode	Heavy Rail	Light Rail	BRT	Bus
Ave. Ridership per trip ( $r_i$ )	400	125	60	40
Ave. Cost per trip ( $c_i$ )	200	80	40	30

- Give a formulation of the problem to maximize the total number of riders the agency services given a fixed daily budget of \$5,000.

# Transit Ridership Formulation

## 1. Decision Variables?

- $X_1$  = number of trips made using heavy rail
- $X_2$  = number of trips made using light rail
- $X_3$  = number of trips made using bus rapid transit (BRT)
- $X_4$  = number of trips made using bus

## 2. Objective Function?

- MAX (Total Ridership)
- Ridership =  $400 * X_1 + 125 * X_2 + 60 * X_3 + 40 * X_4$

## 3. Constraints?

- Cost budget
- Cost =  $200 * X_1 + 80 * X_2 + 40 * X_3 + 30 * X_4$

# Transit Ridership Model

$$MAX(400 * X_1 + 125 * X_2 + 60 * X_3 + 40 * X_4)$$

s.t.

$$200 * X_1 + 80 * X_2 + 40 * X_3 + 30 * X_4 \leq 5000$$

$$X_1, X_2, X_3, X_4 \geq 0$$

## Generalization:

- $M$  = set of modes
- $r_i$  = average ridership per trip for mode  $i$
- $c_i$  = average cost per trip for mode  $i$

$$MAX\left(\sum_{i \in M} r_i X_i\right)$$

s.t.

$$\sum_{i \in M} c_i X_i \leq 5000$$

$$X_i \in R^+, \forall i \in M$$

# Writing the model in OPL Studio

The screenshot displays the OPL Studio interface. The main window shows the following OPL model code:

```
/* Transit Ridership Model */  
range M [1..4];  
  
/* Enter data ridership and cost*/  
int r[M]=[400, 125, 60, 40];  
int c[M]=[200, 80, 40, 30];  
  
/* Define variable as a positive float*/  
var float+ x[M];  
  
maximize sum(i in M) r[i]*x[i]  
subject to  
    sum(i in M) c[i]*x[i] <= 5000;  
};
```

The console window at the bottom shows the optimal solution:

```
Optimal Solution with Objective Value: 10000.0000  
x[1] = 25.0000  
x[2] = 0.0000  
x[3] = 0.0000  
x[4] = 0.0000
```

The status bar at the bottom indicates "Waiting" and shows the system clock as 4:13 PM.

# Transit Ridership: Additional Constraints

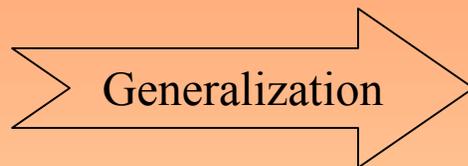
- The agency wants to provide a minimum number of trips  $m_i$ , for each mode  $i$

$$X_1 \geq m_1$$

$$X_2 \geq m_2$$

$$X_3 \geq m_3$$

$$X_4 \geq m_4$$



$$X_i \geq m_i, \forall i \in M$$

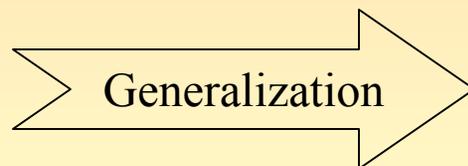
- The agency wants to provide service to a minimum number of riders  $b_i$ , for each mode  $i$ .

$$r_1 X_1 \geq b_1$$

$$r_2 X_2 \geq b_2$$

$$r_3 X_3 \geq b_3$$

$$r_4 X_4 \geq b_4$$



$$r_i X_i \geq b_i, \forall i \in M$$

# Additional Constraints in OPL Studio

The screenshot displays the OPL Studio interface. The main window shows the following OPL model code:

```
/* Transit Ridership Model */  
range M [1..4];  
  
/* Enter data ridership and cost*/  
int r[M]=[400, 125, 60, 40];  
int c[M]=[200, 80, 40, 30];  
  
/*Define mi and bi*/  
int m[M]=[1,2,3,4];  
int b[M]=[50,20,60,40];  
  
/* Define variable as a positive float*/  
var float+ x[M];  
  
maximize sum(i in M) r[i]*x[i]  
subject to{  
    sum(i in M) c[i]*x[i] <= 5000;  
    /* New constraints */  
    forall(i in M) x[i]>=m[i];  
    forall(i in M) r[i]*x[i]>=b[i];  
};
```

The console window at the bottom shows the optimal solution:

```
Optimal Solution with Objective Value: 9790.0000  
x[1] = 23.0000  
x[2] = 2.0000  
x[3] = 3.0000  
x[4] = 4.0000
```

The status bar at the bottom indicates the next solution is being waited for.

# Standard Form of a LP

$$\text{MIN } (c_1 x_1 + c_2 x_2 + \dots + c_n x_n)$$

*s.t.*

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1$$

$$a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2$$

.....

$$a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m$$

$$x_i \geq 0$$

## Any LP can be reduced to its standard form:

- Inequality constraints can be transformed into equality by adding « slack » variables
  - Max problem can be transformed into a MIN problem by reversing signs of objective function coefficients
  - Free variables can be eliminated by replacing them by  $x_j^+ - x_j^-$ , where  $x_j^+$  and  $x_j^-$  are new variables such that  $x_j^+ \geq 0$  and  $x_j^- \geq 0$
- => General problem can be transformed into standard form => only need to develop methods capable of solving standard form problems.

# Linearity

- In a LP, objective AND constraints MUST BE linear
- $\text{MAX}\{x_1, x_2, \dots\}$ ,  $x_i * y_i$ ,  $|x_i|$ , etc  $\Rightarrow$  non-linear if  $x_i$  and  $y_i$  are variables
  - Sometimes there is a way to convert these types of constraints into linear constraints by adding some decision variables
  - Examples:

# Dealing with absolute values

## Example

$$\text{Minimize } .5Y + 2|Z|$$

s.t

$$Y + Z \geq 9$$

=> OPTION 1:

$$|Z| = \text{MAX}\{Z, -Z\}$$

Replace by  $V \geq Z$  and  $V \geq -Z$



*Formulation..1*

$$\text{Minimize } .5Y + 2V$$

s.t.

$$V \geq Z$$

$$V \geq -Z$$

$$Y + Z \geq 9$$

=> OPTION 2

Introduce new variables  $Z^+, Z^-$  such that:

$$Z^+, Z^- \geq 0 \text{ and } Z = Z^+ - Z^-$$



We want  $Z = Z^+$  or  $Z = -Z^-$ , depending on sign of  $Z$

Then,  $Z = Z^+ - Z^-$  and  $|Z| = Z^+ + Z^-$

*Formulation..2*

$$\text{Minimize } .5Y + 2Z^+ + 2Z^-$$

s.t.

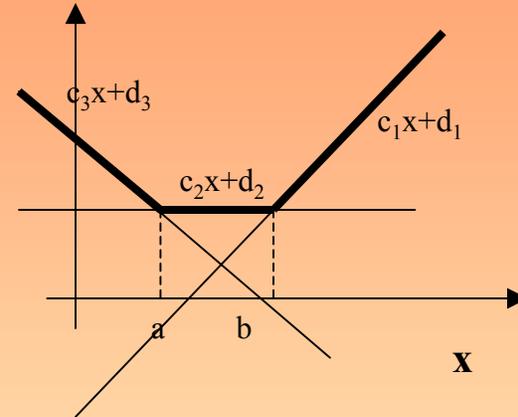
$$Y + Z^+ - Z^- \geq 9$$

$$Z^+, Z^- \geq 0$$

# Dealing with minimizing piece-wise linear convex cost functions

## Example

Cost such that:

$$\begin{cases} c(x) = c_3 \cdot x + d_3, \forall x \in \{-\infty; a\} \\ c(x) = c_2 \cdot x + d_2, \forall x \in \{a; b\} \\ c(x) = c_1 \cdot x + d_1, \forall x \in \{b; +\infty\} \end{cases}$$


What to do?

- Introduce a new variable  $T$  such that:  $T = \text{MAX} \{c_3x + d_3, c_2x + d_2, c_1x + d_1\}$

- In linear form:

$$T \geq c_3x + d_3$$

$$T \geq c_2x + d_2$$

$$T \geq c_1x + d_1$$

# Example

A marketing manager has an advertising budget of \$150,000. In order to increase automobile sales, the firm is considering advertising in newspapers and on TV. The more a particular medium is used, the less effective is each additional ad. Each newspaper ad costs \$1,000 and each TV ad costs \$10,000. At most 30 newspaper ads and at most 15 TV ads can be placed.

	No. of Ads	New Customers
Newspaper	1-10	900
	11-20	600
	21-30	300
TV	1-5	10,000
	6-10	5,000
	11-15	2,000

# Problem Formulation 1

$x_{N1}$  : # of newspaper ads placed between 1-10

$x_{N2}$  : # of newspaper ads placed between 11-20

$x_{N3}$  : # of newspaper ads placed between 21-30

$x_{T1}$  : # of TV ads placed between 1-5

$x_{T2}$  : # of TV ads placed between 6-10

$x_{T3}$  : # of TV ads placed between 11-15

$$\text{MAX}(900x_{N1} + 600x_{N2} + 300x_{N3} + 10,000x_{T1} + 5000x_{T2} + 2000x_{T3})$$

*s.t*

$$1000 * (x_{N1} + x_{N2} + x_{N3}) + 10000 * (x_{T1} + x_{T2} + x_{T3}) \leq 150,000$$

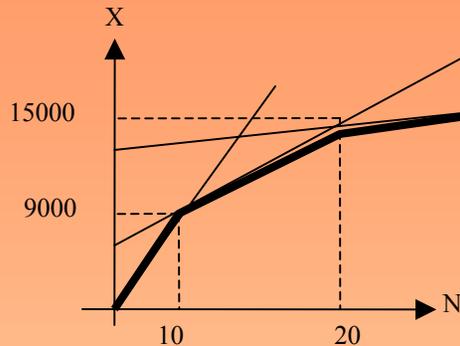
$$0 \leq x_{N1}, x_{N2}, x_{N3} \leq 10$$

$$0 \leq x_{T1}, x_{T2}, x_{T3} \leq 5$$

$$x_{N1}, x_{N2}, x_{N3}, x_{T1}, x_{T2}, x_{T3} \in \mathbb{Z}^+$$

What does this formulation rely on?

# Alternative Formulation 2

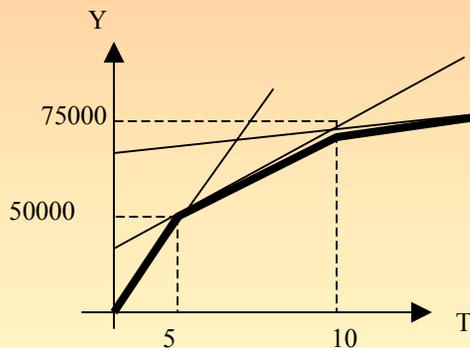


$$N \in \{1, \dots, 10\} \Rightarrow X = 900N$$

$$N \in \{11, \dots, 20\} \Rightarrow X = 900 \cdot 10 + 600(N - 10) = 3000 + 600N$$

$$N \in \{21, \dots, 30\} \Rightarrow X = 900 \cdot 10 + 600 \cdot 10 + 300(N - 20) = 9000 + 300N$$

$$\text{Therefore } X = \text{MIN}\{900N, 3000 + 600N, 9000 + 300N\}$$



$$T \in \{1, \dots, 5\} \Rightarrow Y = 10000T$$

$$T \in \{6, \dots, 10\} \Rightarrow Y = 10000 \cdot 5 + 5000(T - 5) = 25000 + 5000T$$

$$T \in \{11, \dots, 15\} \Rightarrow Y = 10000 \cdot 5 + 5000 \cdot 5 + 2000(T - 10) = 55000 + 2000T$$

$$\text{Therefore } Y = \text{MIN}\{10000T, 25000 + 5000T, 55000 + 2000T\}$$

Variables:

X: number of customers reached via newspaper ads

Y: number of customers reached via TV ads

N: number of newspaper ads

T: number of TV ads

X and Y are piecewise linear functions of N and T respectively

*Maximize*  $(X + Y)$

*s.t.*

$$1000N + 10000T \leq 150000$$

$$X \leq 900N$$

$$X \leq 3000 + 600N$$

$$X \leq 9000 + 300N$$

$$Y \leq 10000T$$

$$Y \leq 25000 + 5000T$$

$$Y \leq 55000 + 2000T$$

$$N \leq 30$$

$$T \leq 15$$

12/31/2003

Barnhart 1.224J

20

# Alternative Formulation 3

$x_N$  : # of newspaper ads

$x_T$  : # of TV ads placed

$y_i$  : supplementary variables,  $i=1,2,\dots,6$

$$\text{MAX}(9000 - 900 * y_1 + 6000 - 600 * y_2 + 3000 - 300 * y_3 + 50000 - 10,000 * y_4 + 25,000 \\ - 5,000 * y_5 + 10,000 - 2,000 * y_6)$$

*s.t*

$$y_1 \geq 10 - x_N$$

$$y_2 \geq 20 - x_N - y_1$$

$$y_3 \geq 30 - x_N - y_1 - y_2$$

$$y_4 \geq 5 - x_T$$

$$y_5 \geq 10 - x_T - y_4$$

$$y_6 \geq 15 - x_T - y_4 - y_5$$

$$x_N \leq 30$$

$$x_T \leq 15$$

$$1000 * x_N + 10000 * x_T \leq 150000$$

$$x_N, x_T \in \mathbb{Z}^+$$

$$y_1, y_2, y_3, y_4, y_5, y_6 \geq 0$$

# Additional Constraints

## How would you model the following?

- (1) At most 30 ads can be placed in total
- (2) There is a 20% discount for each additional TV ad if the number of TV ads exceeds 12

- (1) Add the following constraint:

$$x_N + x_T \leq 30$$

- (2) Define a new supplementary variable  $y_d$  and replace  $y_6$  with  $y'_6 + y_d$  in previous formulation, and add:

$$1000 * x_N + 10000 * x_T - 2000 * (3 - y'_6) \leq 150000$$

$$12 - x_T - y_4 - y_5 \leq y_d$$

$$y_d \geq 0$$

# Formulating the Model

- Multiple ways to develop a model formulation

- 1- Decide on an initial set of decision variables

- Traditionally letters from the end of the alphabet, use of subscripts, ordering of subscripts

- 2- Determine objective function:

- obtainable from problem statement
- Can be very complex

- 3- Determine the constraints:

- Variable-value constraints: non-negativity, binary constraints
- Capacity constraints, demand constraints, balance flow constraints
- Sometimes necessitates introduction of additional variables

# **Solving the LP**

# Outline

## 2. Solving linear programs

- Linear Programs: Forms and Notation
- Basic/Non-basic variables
- Dual Variables
- Reduced Costs
- Optimality Conditions
- Example
- Simplex algorithm
- Sensitivity Analysis
  - Introduction of a new variable
  - Addition of a new constraint
  - Change in the cost coefficient of a non-basic variable
  - Change in the constraint coefficient of a non-basic variable

# Linear programs: Form and notation

$$\text{Min } (c_1x_1 + c_2x_2 + \dots + c_nx_n)$$

*s.t*

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

:

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

$$x_i \geq 0, \forall i$$

$$\text{Min. } c'x$$

*s.t*

$$Ax = b$$

$$x \geq 0$$

- $c$  in an  $n \times 1$  vector
- $x$  is an  $n \times 1$  vector
- $A$  is an  $m \times n$  matrix
- $b$  is an  $m \times 1$  vector 26

# Basic Solutions

- If an LP has an optimal solution, it must also have an optimal basic solution
  - A basic solution is one in which all but  $m$  variables take on value zero
    - $n - m$  non-basic variables
  - These  $m$  variables are referred to as basic variables (note that basic variables can also take on value 0)

# Dual Variables

- Let  $\pi$  be the  $m \times 1$  vector of dual variables associated with the  $m$  constraints
- Given a basic solution, the dual variable value of a constraint can be interpreted as the value of relaxing the constraint by one unit.
  - If the constraint is not binding, the dual value is equal to 0 and relaxing it by one unit has no effect on the optimal solution

# Reduced Costs

The reduced cost of variable  $x_i$  is:

$$c_i - A_i' \Pi, \text{ or}$$

$$c_i - a_{1i} \Pi_1 - a_{2i} \Pi_2 - \dots - a_{mi} \Pi_m$$

Reduced costs of a variable  $x_i$  can be viewed as an estimate of the change in the objective function value achieved by increasing  $x_i$  by one unit.

# Calculating reduced costs

	X1	X2	X3	X4	X5	X6	b	Dual
Cost(j)	-10	-12	-12	0	0	0		
A(i,j)	1	2	2	1	0	0	20	-3.6
	2	1	2	0	1	0	20	-1.6
	2	2	1	0	0	1	20	-1.6

$$\text{Red. Cost } X(j) = C(j) - A(1,j) * \pi_1 - A(2,j) * \pi_2 - A(3,j) * \pi_3$$

$$\text{Red. Cost } X(1) = -10 - 1 * (-3.6) - 2 * (-1.6) - 2 * (-1.6) = 0$$

$$\text{Red. Cost } X(2) = -12 - 2 * (-3.6) - 1 * (-1.6) - 2 * (-1.6) = 0$$

$$\text{Red. Cost } X(3) = -12 - 2 * (-3.6) - 2 * (-1.6) - 1 * (-1.6) = 0$$

$$\text{Red. Cost } X(4) = 0 - 1 * (-3.6) - 0 - 0 = 3.6$$

$$\text{Red. Cost } X(5) = 0 - 0 - 1 * (-1.6) - 0 = 1.6$$

$$\text{Red. Cost } X(6) = 0 - 0 - 0 - 1 * (-1.6) = 1.6$$

# Solving the LP

- Many algorithms can be used to solve the LP
- Simplex algorithm (most popular)
  - Searches for an optimal solution by moving from one basic solution to another, along the edges of the feasible polygon, in direction of cost decrease (Graphically, moves from corner to corner)
- Interior Point Methods (more recent)
  - Approaches the situation through the interior of the convex polygon
  - Affine Scaling
  - Log Barrier Methods
  - Primal-dual methods

# The Simplex algorithm for minimization problems

- 1- Compute the reduced costs of all non-basic variables. If they are all non-negative, stop.
- 2- If not, choose some non-basic variable with negative reduced cost.
- 3- Identify an active variable to remove from the basis.
- 4- Solve for the value of the new set of basic variables.
- 5- Solve for the new value of the dual variables.
- 6- Return to Step 1

# Simplex Optimality Conditions (for minimization problems)

The current feasible solution  $x$  is optimal when:

- The reduced costs of all basic variables equal 0
  - Maintained at each iteration of the simplex algorithm
- The reduced costs of all non-basic variables are non-negative
  - Not maintained at each iteration of the simplex algorithm
- Dual variables are feasible for the dual problem
- Complementary slackness is satisfied (maintained at each iteration of the simplex algorithm)
  - Dual variable value is zero unless its associated constraint is binding (has zero slack)
  - Value of the decision variable  $x_i$  is zero unless its associated reduced cost is zero
    - $x_i$  is non-zero only if its associated reduced cost is zero.

## Example\*

A company produces 3 products. Each unit of product 1,2, and 3 generate a profit of \$10, \$12 , and \$12 respectively.

Each product has to go through a manufacturing, assembly, and testing phase. The company's resources are such that only 20 hours of manufacturing, 20 hours of assembly, and 20 hours of testing are available. Each unit of product 1 has to spend 1 hr in manufacturing, 2 hrs in assembly, and 2 hrs in testing. Each unit of product 2 has to spend 2 hrs in manufacturing, 1 hr in assembly, and 2 hrs in testing. Each unit of product 3 has to spend 2 hrs in manufacturing, 2 hrs in assembly, and 1 hr in testing. Company ABC wants to know how many units of each product it should produce, in order to maximize its profit.

$$\text{Maximize}(10X_1 + 12X_2 + 12X_3)$$

*s.t.*

$$X_1 + 2X_2 + 2X_3 \leq 20$$

$$2X_1 + X_2 + 2X_3 \leq 20$$

$$2X_1 + 2X_2 + X_3 \leq 20$$

$$X_1, X_2, X_3 \geq 0$$

### In Standard Form

$$\text{Minimize}(-10X_1 - 12X_2 - 12X_3)$$

*s.t.*

$$X_1 + 2X_2 + 2X_3 + X_4 = 20$$

$$2X_1 + X_2 + 2X_3 + X_5 = 20$$

$$2X_1 + 2X_2 + X_3 + X_6 = 20$$

$$X_1, X_2, X_3, X_4, X_5, X_6 \geq 0$$

\*Source: Optimization Methods p101

# In OPL Studio...

The screenshot displays the OPL Studio interface. The main window shows a linear programming model with the following code:

```
/*Example*/  
constraint ct[1..3];  
range M[1..3];  
  
/* Enter coefficients*/  
int A[1..3,1..3]= [[1,2,2], [2,1,2], [2,2,1]];  
int C[1..3]=[-10,-12,-12];  
  
/* Define variable as a positive float*/  
var float+ X[M];  
  
minimize sum(j in 1..3) C[j]*X[j]  
subject to{  
forall (i in 1..3)  
ct[i]: sum(j in 1..3) A[i,j]*X[j] <=20;  
};  
display (j in 1..3) X[j];  
display (j in 1..3) X[j].rc; /* display reduced costs*/  
display (i in 1..3) ct[i].dual; /*display dual values*/
```

The console window at the bottom shows the optimal solution:

```
Optimal Solution with Objective Value: -136.0000  
X[1] = 4.0000  
X[2] = 4.0000  
X[3] = 4.0000  
  
X[1].rc = 0.0000  
X[2].rc = 0.0000  
X[3].rc = 0.0000  
  
ct[1].dual = -3.6000  
ct[2].dual = -1.6000  
ct[3].dual = -1.6000
```

A red circle highlights the solution values in the console window. The status bar at the bottom indicates the current state is "Waiting".

# In OPL With Slack variables

The screenshot displays the OPL Studio interface. The main window shows the OPL model code, and the console window shows the optimal solution. A pink oval highlights the solution output in the console.

```
/*Example with slack variables*/
constraint ct[1..3];
/* Enter coefficients*/
int A[1..3,1..6]= [[1,2,2,1,0,0], [2,1,2,0,1,0], [2,2,1,0,0,1]];
int C[1..6]=[-10,-12,-12,0,0,0];
/* Define variable as a positive float*/
var float+ X[1..6];

minimize sum(j in 1..6) C[j]*X[j]
subject to{
forall (i in 1..3)
ct[i]: sum(j in 1..6) A[i,j]*X[j] =20;
};
display (j in 1..6) X[j];
display (j in 1..6) X[j].rc; /* display reduced costs*/
display (i in 1..3) ct[i].dual; /*display dual values*/
```

Optimal Solution with Objective Value: -136.0000

X[1] = 4.0000  
X[2] = 4.0000  
X[3] = 4.0000  
X[4] = 0.0000  
X[5] = 0.0000  
X[6] = 0.0000

X[1].rc = 0.0000  
X[2].rc = 0.0000  
X[3].rc = 0.0000  
X[4].rc = 3.6000  
X[5].rc = 1.6000  
X[6].rc = 1.6000

ct[1].dual = -3.6000  
ct[2].dual = -1.6000  
ct[3].dual = -1.6000

Next solution? ..\..\..\Documents and Settings\Yasmine El Aj\My Docume Waiting

Start | WebMail :: ... | IP\_OVERVI... | NETWORK ... | LPintro.ppt | LP2\_solvin... | OPL Studi... | OPL online ... | 1:47 PM

# Example continued

## Basic vs. non-basic variables

- 3 constraints  $\Rightarrow$  (at most) 3 variables are basic variables
- $X_1, X_2, X_3 \geq 0 \Rightarrow X_1, X_2, X_3$  are basic
- Non-basic variables  $\Rightarrow X_4, X_5, X_6 = 0$

## Dual Values

- All dual values different from 0  $\Rightarrow$  All constraints are binding
- Dual Value (1) = -3.6  $\Rightarrow$  Relaxing constraint 1 by 1 unit (right hand side equal to 21 instead of 20) would result in a decrease of 3.6 in the objective value.
- Dual Value (2) = Dual value (3) = -1.6  $\Rightarrow$  Relaxing constraint 2 or constraint 3 by 1 unit would result in a decrease of 1.6 in the objective value.

## Reduced Costs

- Reduced costs of  $X_1, X_2, X_3 = 0 \Rightarrow$  reduced costs of basic variables equal 0
- Reduced costs of  $X_4, X_5, X_6$  (non-basic variables)  $\geq 0$
- Solution is optimal because all reduced costs (basic + non-basic) are  $\geq 0$  and we are solving a minimization problem

$\Rightarrow$  **Complementary slackness is satisfied**

Basic Variables		
	Value	Red. Cost
X1	4	0
X2	4	0
X3	4	0

Slack Variables		Const.
	Value	Dual
X4	0	-3.6
X5	0	-1.6
X6	0	-1.6

# Calculating reduced costs

	X1	X2	X3	X4	X5	X6	b	Dual
Cost(j)	-10	-12	-12	0	0	0		
A(i,j)	1	2	2	1	0	0	20	-3.6
	2	1	2	0	1	0	20	-1.6
	2	2	1	0	0	1	20	-1.6

$$\text{Red. Cost } X(j) = C(j) - A(1,j) * \pi_1 - A(2,j) * \pi_2 - A(3,j) * \pi_3$$

$$\text{Red. Cost } X(1) = -10 - 1 * (-3.6) - 2 * (-1.6) - 2 * (-1.6) = 0$$

$$\text{Red. Cost } X(2) = -12 - 2 * (-3.6) - 1 * (-1.6) - 2 * (-1.6) = 0$$

$$\text{Red. Cost } X(3) = -12 - 2 * (-3.6) - 2 * (-1.6) - 1 * (-1.6) = 0$$

$$\text{Red. Cost } X(4) = 0 - 1 * (-3.6) - 0 - 0 = 3.6$$

$$\text{Red. Cost } X(5) = 0 - 0 - 1 * (-1.6) - 0 = 1.6$$

$$\text{Red. Cost } X(6) = 0 - 0 - 0 - 1 * (-1.6) = 1.6$$

# SENSITIVITY ANALYSIS

# Local Sensitivity Analysis

- How does the objective function value and optimality conditions change when:
  - A new variable is introduced
  - A new inequality is introduced
  - The cost coefficient of a non-basic variable changes
  - The constraint coefficient of a non-basic variable changes

# Introduction of a new variable

- Feasibility of the current solution is not affected
- Need to check if current solution is still optimal (i.e. all reduced costs  $\geq 0$ )
- Calculate the reduced cost of the new variable

$$C_{new} - \sum_{i \in M} a_{i,new} \pi_i$$

- If the reduced cost  $\geq 0$ , the current solution remains optimal
- If the reduced cost  $< 0$ , the current solution is no longer optimal. The new variable enters the basis at the next iteration of the Simplex method.

# Simplex Optimality Conditions (for minimization problems)

The current feasible solution  $x$  is optimal when:

- The reduced costs of all basic variables equal 0
  - Maintained at each iteration of the simplex algorithm
- The reduced costs of all non-basic variables are non-negative
  - Not maintained at each iteration of the simplex algorithm
- Dual variables are feasible for the dual problem
- Complementary slackness is satisfied (maintained at each iteration of the simplex algorithm)
  - Dual variable value is zero unless its associated constraint is binding (has zero slack)
  - Value of the decision variable  $x_i$  is zero unless its associated reduced cost is zero
    - $x_i$  is non-zero only if its associated reduced cost is zero.

# Example 1:

- Company ABC is thinking about introducing a new product. The new product would generate a profit of \$11/ unit. It would require 2 hours of manufacturing, 2 hrs of assembly, and 2 hrs of testing. Should Company ABC introduce it?
- Calculate reduced cost of new product:
- Red. Cost (New) =  $-11 - 2*(-3.6) - 2*(-1.6) - 2*(-1.6)$   
= 2.6
- Red. Cost (New)  $\geq 0 \Rightarrow$  do NOT introduce the product

## Example 2

- What if the new product generated a profit of \$14 instead of \$11?
  - Red. Cost (New) =  $-14 - 2*(-3.6) - 2*(-1.6) - 2*(-1.6) = -0.4$
  - Red. Cost (New)  $\leq 0 \Rightarrow$  Solution could be improved by introducing the new product.
- $\Rightarrow$  Re-solve the problem to get the new optimal solution

# A new inequality constraint is added

- If current solution satisfies the new constraint, the current solution is optimal
- Otherwise, re-solve

# Change in the cost coefficient of a non-basic variable $X_v$

- $C_v$  becomes  $C_v + \delta$ , with  $\delta \geq 0$  or  $\delta \leq 0$
- Feasibility of current solution not affected
- Check optimality conditions
- The only reduced cost affected is that of the variable for which the coefficient was modified
  - Let  $\check{C}_v$  be the current reduced cost
  - New reduced cost:  $C_v + \delta - \sum_{i \in M} a_{i,v} \pi_i = \check{C}_v + \delta$
  - If  $\check{C}_v \geq -\delta \Rightarrow$  current solution is still optimal
  - If  $\check{C}_v < -\delta \Rightarrow$  current solution is no longer optimal

# Example:

$$\text{Minimize}(-5X_1 - X_2 + 12X_3)$$

*s.t.*

$$3X_1 + 2X_2 + X_3 = 10$$

$$5X_1 + 3X_2 + X_4 = 16$$

$$X_1, X_2, X_3, X_4 \geq 0$$

OPL Studio - [..\..\..\Documents and Settings\Yasmine El Alj\My Documents\2003RESEARCH\OPL\_Examples\exple\_modcost.mod]

File Edit View Project Execution Debug Options Window Help

exple\_solving.mod  
exple\_solving\_slack.nr  
exple\_solving\_newval  
exple\_modcost.mod

```
/* Example */  
constraint cst[1..2];  
  
/* Enter coefficients*/  
int A[1..2,1..4]= [[3,2,1,0], [5,3,0,1]];  
int C[1..4]=[-5,-1,12,0];  
int b[1..2]=[10,16];  
  
/* Define variable as a positive float*/  
var float+ X[1..4];  
  
minimize sum(j in 1..4) C[j]*X[j]  
subject to{  
forall (i in 1..2)  
cst[i]: sum(j in 1..3) A[i,j]*X[j] =b[i];  
};  
display (j in 1..4) X[j];  
display (j in 1..4) X[j].rc; /* display reduced costs*/  
display (i in 1..2) cst[i].dual; /*display dual values*/
```

Model Projects

Optimal Solution with Objective Value: -12.0000  
X[1] = 2.0000  
X[2] = 2.0000  
X[3] = 0.0000  
X[4] = 0.0000  
  
X[1].rc = 0.0000  
X[2].rc = 0.0000  
X[3].rc = 2.0000  
X[4].rc = 0.0000  
  
cst[1].dual = 10.0000  
cst[2].dual = -7.0000

Console Solutions Optimization Log Solver CPLEX

Next solution? ..\..\..\Documents and Settings\Yasmine El Alj\My Docume Waiting

Start We... IP... NET... LPin... LP2... OPL... OPL... Micr... SEN... 3:52 PM

# Example: Change in cost coefficient

Current reduced cost of  $X_3=2$ ,  $\pi_1= 10$ ,  $\pi_2= -7$

Change cost of  $X_3$  from 12 to 11 ( $\delta= -1$ )

- $2 \geq 1 \Rightarrow$  current solution is optimal
- New Red. Cost ( $X_3$ ) =  $11 - 1 * 10 - 0 = 1$

Change cost of  $X_3$  from 12 to 6 ( $\delta= -6$ )

- $2 \leq 6 \Rightarrow$  current solution no longer optimal
- New Red. Cost ( $X_3$ ) =  $6 - 1*10 - 0 = -4$
- Red. Cost ( $X_3$ )  $\leq 0 \Rightarrow X_3$  will become a basic variable  $\Rightarrow$  Re-solve

# Change in the constraint coefficient of a non-basic variable (the $v^{\text{th}}$ variable)

- Feasibility conditions not affected
- Check optimality conditions
- Only the reduced cost of the  $v^{\text{th}}$  column is affected
- Change of the coefficient on the  $z^{\text{th}}$  row and  $v^{\text{th}}$  column by  $\alpha$  ( $\alpha \geq 0$  or  $\alpha \leq 0$ )
  - Current reduced cost =  $\check{C}_v$
  - New reduced cost  $C_v - \sum_{i \neq z} a_{i,v} \pi_i - (a_{z,v} + \alpha) \pi_z = \check{C}_v - \alpha \pi_z$
  - If  $\alpha \leq \check{C}_v / \pi_z \Rightarrow$  current solution remains optimal
  - If  $\alpha > \check{C}_v / \pi_z \Rightarrow$  current solution is no longer optimal

# Example: Change in constraint coefficient of a non-basic variable

Current reduced cost of  $X_3=2$ ;  $\pi_1= 10$ ,  $\pi_2= -7$

Change coefficient in constraint 1 from 1 to -1 ( $\alpha= -2$ )

- $-2 \leq 2/10 \Rightarrow$  current solution is optimal
- New Red. Cost ( $X_3$ ) =  $12 - (-1) * 10 = 22$

Change coefficient in constraint 1 from 1 to 2 ( $\alpha=1$ )

- $1 > 2/10 \Rightarrow$  current solution no longer optimal
- New Red. Cost ( $X_3$ ) =  $12 - 2*10 - 0 = -8$
- Red. Cost ( $X_3$ )  $\leq 0 \Rightarrow X_3$  will become a basic variable  $\Rightarrow$  Re-solve