Optimization Background for Network Design

David Tipper
Associate Professor
Graduate Telecommunications and Networking Program
University of Pittsburgh
tipper@tele.pitt.edu

Slides 5
http://www.sis.pitt.edu/~dtipper/2110.html

Optimization Review

- Algorithms for Logical Layer Design
  - (Graph Theory, Optimization)
- Optimization Techniques
  - Seek to find best (maximum or minimum) solution as determined by an *objective function* *f(X)*
  - Set of unknown *decision variables* *X*
  - *Constraints* limit the possible values for the variables
  - *Feasible space* is the set of solutions that satisfy the constraints

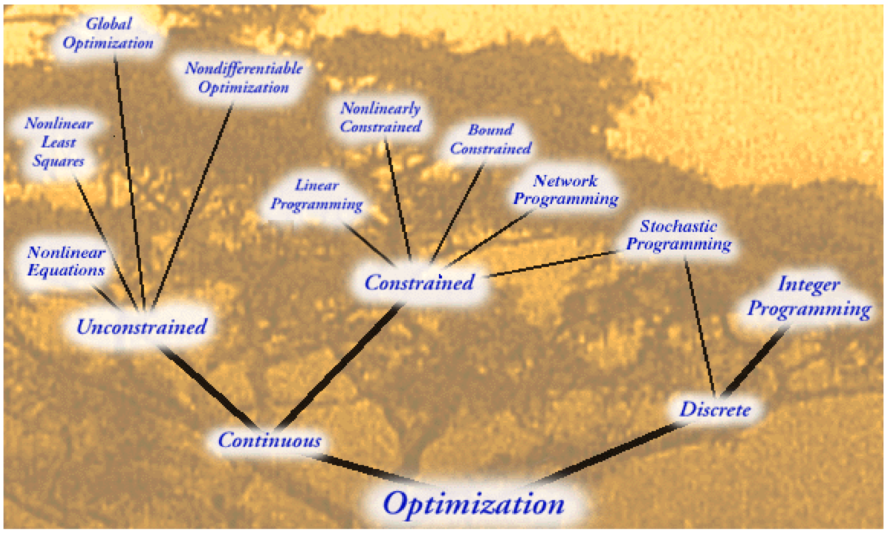
$$\min_{X} f(X)$$

$$\text{subject to } X \in A$$

- Definition
  - "A Mathematical Programming Model is a mathematical decision model for planning (programming) decisions that optimize an objective function and satisfy limitations imposed by mathematical constraints." **1

**1 T.W. Knowles, Management Science: Building and Using Models, Irwin, 1989.

Types of Optimization Problems



Constrained Optimization

- General Symbolic Model

Maximize (or minimize): $f(x_1, x_2 \dots x_n)$ } Objective

Subject to:
$$\left.\begin{array}{lll} g_1(x_1, x_2 \dots x_n) & \{\le, \ge, =\} & b_1 \\ g_2(x_1, x_2 \dots x_n) & \{\le, \ge, =\} & b_2 \\ \vdots & & \\ g_m(x_1, x_2 \dots x_n) & \{\le, \ge, =\} & b_m \end{array}\right\} \text{Constraints}$$

… where $x_1, x_2 \dots x_n$ are the ***decision variables***

- Solution methods
  - **brute-force, analytical and heuristic solutions**
  - **linear/integer/convex programming**

Mathematical Programming

- Types of Mathematical Programs:
  - **Linear Programs (LP):** the objective and constraint functions are linear and the decision variables are continuous.
  - **Integer (Linear) Programs (ILP):** one or more of the decision variables are restricted to integer values only and the functions are linear.
    - Pure IP: all decision variables are integer.
    - Mixed IP (MIP): some decision variables are integer, others are continuous.
    - 1/0 MIP: some or all decision variables are further restricted to be valued either "1" or "0" that is Binary Variables.
  - **Nonlinear Programs:** one or more of the functions is not linear.

Linear Programming

- General Symbolic Form

Maximize: $c_1x_1 + c_2x_2 + \ldots c_nx_n$ } Objective

Subject to:

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \quad \{\le,\ge,=\} \quad b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \quad \{\le,\ge,=\} \quad b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \quad \{\le,\ge,=\} \quad b_m \end{array}\right\} \text{Constraints}$$

$0 \le x_j, \quad j = 1, \ldots, n$ } Bounds

… where a_{ij}, b_j, c_j are the model *parameters.*

Linear Programming

- Can be written in matrix formulation

Maximize:	$c^T x$	} Objective
Subject to:	$Ax = b$	} Constraints
	$0 \le x_j \quad \forall j$	} Bounds

...where *c, A, b* are *parameters*

Linear Programming

- General Restrictions
  - All decision variables must be nonnegative $x_j \ge 0$.
  - Constant terms cannot appear on the LHS of a constraint.
  - No variable can appear on the RHS of a constraint.
  - No variable can appear more than once in a function, i.e. objective or constraint.
- Steps for Formulating LP Models
  - Construct a verbal model.
  - Define the decision variables.
  - Construct the math model.
- **Feasible solution** - set of all points satisfying all constraints and sign restrictions
  Optimal solution to an LP – a point in the feasible region with the best objective function value

Formulating LP Problems

- An example[**2]
  - A steel company must decide how to allocate production time on a rolling mill. The mill takes unfinished slabs of steel as input and can produce either of two products: bands and coils.
    The products come off the mill at different rates and also have different profit-abilities:

	Tons/hour	Profit/ton
Bands	200	$25
Coils	140	$30

  - The weekly production that can be justified based on current and forecast orders are:

Maximum tons:	Bands	6,000
	Coils	4,000

[**2] from, R. Fourer, D. Gay, B. Kernighan, AMPL, Boyd & Fraser, 1993, pp. 2-10.

Formulating LP Problems

- Example (cont'd)
  - The question facing the company is:
    If 40 hours of production time are available, how many tons of bands and coils should be produced to bring in the greatest total profit?
- Constructing the Verbal model
  - Put the objective and constraints into words.
  - For constraints, use the form

{a verbal description of the LHS} {a relationship} {an RHS constant}

Maximize:	total profit
Subject to:	total number of production hours ≤ 40
	tons of bands produced ≤ 6,000
	tons of coils produced ≤ 4,000

Formulating LP Problems

- Definine the Decision Variables
  - X_B number of tons of bands produced.
  - X_C number of tons of coils produced.

- Construct the Symbolic Model

Maximize: $25X_B + 30X_C$

Subject to: $(1/200)X_B + (1/140)X_C \le 40$

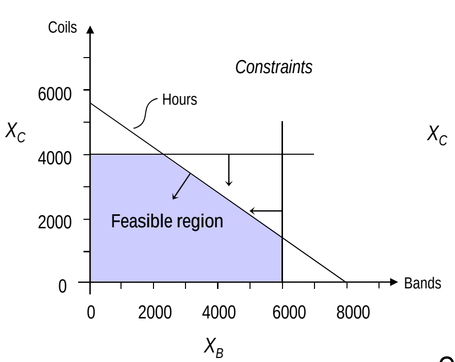
$$0 \le X_B \le 6000$$

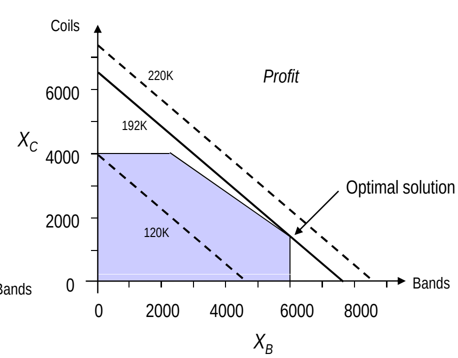
$$0 \le X_C \le 4000$$

For a Two variable problem can solve graphically by ploting constraints and objective function

Solving LP Problems

- Graphical Solution Method





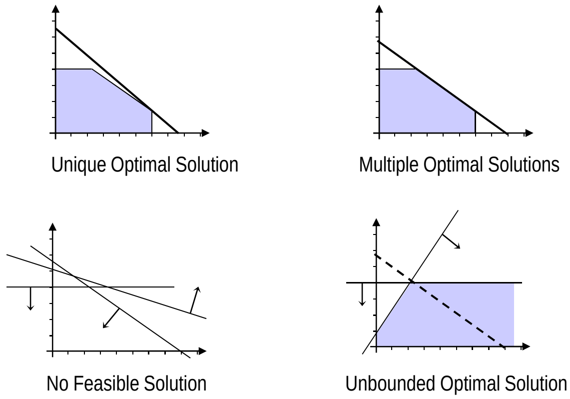
Optimal Solution: $X_B = 6000$, $X_C = 1400$

Profit = 25*6000 + 30*1400

= $192,000

Solving LP Problems

- 4 Possible Outcomes



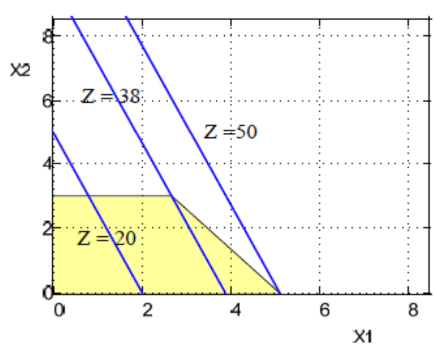
Example 2

Maximize $Z = 10 X_1 + 4 X_2$
Subject to

$$5X_1 + 4 X_2 \leq 25$$
$$X_2 \leq 3$$
$$X_1 \geq 0, \quad X_2 \geq 0$$

The feasible solution space is the area shaded in the figure below.

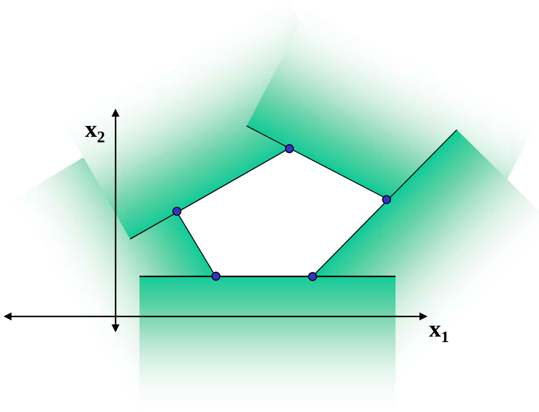
The optimal solution is x1 = 5 and x2 = 0, which gives the maximum objective value z = 50.



Solving Linear Programs

In general problem is too complex for graphical solution

- Simplex method
  - Efficient algorithm to solve LP problems by performing matrix operations on the LP Tableau
  - Developed by George Dantzig (1947)
  - Can be used to solve small LP problems by hand
  - Equivalent to checking corner points



Linear Program in Standard Form

First put in standard form

- **indices**
  - j=1,2,...,n — variables
  - i=1,2,...,m — equality constraints
- **constants**
  - $c = (c_1,c_2,...,c_n)$ — cost coefficients
  - $b = (b_1,b_2,...,b_m)$ — constraint left-hand-sides
  - $A = (a_{ij})$ — m × n matrix of constraint coefficients
- **variables**
  - $x = (x_1, x_2,...,x_n)$

Linear program

- maximize

  $z = \Sigma_{j=1,2,...,n} c_j x_j$
- subject to

  $\Sigma_{j=1,2,...,n} a_{ij} x_j = b_i$, i=1,2,...,m

  $x_j \geq 0$, j=1,2,...,n

Linear program (matrix form)

- maximize

  cx
- subject to

  Ax = b

  x ≥ 0

n > m

rank(**A**) = m

Simplex Method

- Slack Variables

  add a slack variable x_{n+i} to make constraint an equality

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad change\ \ to \quad \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i \ , \ x_{n+i} \ge 0$$

$$\sum_{j=1}^{n} a_{ij} x_j \ge b_i \quad change\ \ to \quad \sum_{j=1}^{n} a_{ij} x_j - x_{n+i} = b_i \ , \ x_{n+i} \ge 0$$

- Nonnegative unconstrained Variables

  x_n unconstrained $x_n - x_{n+i} + x_{n+i+1} = 0 \quad x_{n+i} \ge 0 \, , \ x_{n+i+1} \ge 0$

<u>Exercise</u>: transform the following LP to the standard form

Maximize: z = $x_1 + x_2$

subject to $2x_1 + 3x_2 \le 6$

$x_1 + 7x_2 \ge 4$

$x_1 + x_2 = 3$

$x_1 \ge 0 \, , \ x_2 \ge 0$

Simplex Method

<u>Exercise</u>: transform the following LP to the standard form

Maximize: z = $x_1 + x_2$

subject to

$2x_1 + 3x_2 + x_3 = 6$

$x_1 + 7x_2 - x_4 = 4$

$x_1 + x_2 = 3$

$x_1 \ge 0 \, , \ x_2 \ge 0 \, , \ x_3 \ge 0 \, , \ x_4 \ge 0$

Some software tools require converting max to min by multiplying by -1

For example Min –z is same as Max z , that is

Min $- x_1 - x_2$ is the same as Max above

Simplex Method

- After LP is in standard form
- Find a basic feasible solution (maybe slack variable with base variable set to zero), move from corner to corner via swapping columns and eliminating slack variables.
- Algorithm
  1. Find a basic feasible solution and form tableau
  2. Repeat
     1. If all coefficients in objective row => 0 stop
     2. Else, pick column with most negative coefficient
     3. Pick row with least positive ratio of rhs/(column value)
     4. Normalize Row so pivot value =one
     5. Use Gaussian elimination to remove make rest of column zero

LP Example

- Two types of leather belts: deluxe and regular, each requires 1 sq. yard of leather
- Each week: only 40 sq yard leather & 60 hrs of labor skill available
- Regular belt: 1 hr labor -> $3 profit
  Deluxe belt: 2 hr labor -> $4 profit
- Variable: x_1 = # deluxe belts produced/wk
  x_2 = # regular belts produced/wk

- LP: maximize profit $z = 4x_1 + 3x_2$

  s.t. $x_1 + x_2 \leq 40$

  $2x_1 + x_2 \leq 60$

  $x_1, x_2 \geq 0$

Simplex method for standard Max problem

- From previous problem:

maximize profit $z = 4x_1 + 3x_2$

subject to

$x_1 + x_2 \leq 40$

$2x_1 + x_2 \leq 60$

$x_1, x_2 \geq 0$

- Step1: Convert LP to standard form
  - Add slack variable to each inequality constraint
  - Turn constraints to equations

$$z - 4x_1 - 3x_2 = 0$$

s.t.

$$x_1 + x_2 + x_3 = 40$$

$$2x_1 + x_2 + x_4 = 60$$

Simplex method for standard Max problem

- Step2: Write down the simplex tableau
  - Start with NBV = {x1, x2} and BV = {z, x3, x4}

z	x1	x2	x3	x4	rhs	BV
1	-4	-3	0	0	0	z = 0
0	1	1	1	0	40	x_3 = 40
0	2	1	0	1	60	x_4 = 60

- Step3: Choose entering variable and do test ratio
  - entering variable: x1 (increase x1 one unit -> z increases 4 units)
  - Pivot row: row2

	z	x1	x2	x3	x4	rhs	BV	ratio
Row 0:	1	-4	-3	0	0	0	z = 0	
Row 1:	0	1	1	1	0	40	x_3 = 40	x_1 <= 40
Row 2:	0	2	1	0	1	60	x_4 = 60	x_1 <= 30

Simplex method for standard Max problem

- Step4: Perform pivoting
  - Make coefficient of x1 in row2 to be one and all other rows to be zeros
  - departing variable: x4 and BV = {z, x3, x1}

z	x1	x2	x3	x4	rhs	BV
1	0	-1	0	2	120	z = 120
0	0	0.5	1	-0.5	10	x_3 = 10
0	1	0.5	0	0.5	30	x_1 = 30

- Repeat step 3-4
  - entering variable: x2 (increase x2 one unit -> z increases one unit)
  - Pivot row: row1

	z	x1	x2	s1	s2	rhs	BV	ratio
Row 0:	1	0	-1	0	2	120	z = 120	
Row 1:	0	0	0.5	1	-0.5	10	x_3 = 10	x_2 <= 20
Row 2:	0	1	0.5	0	0.5	30	x_1 = 30	x_2 <= 60

Simplex method for standard Max problem

- Optimal solution is reached
  - No new entering variable is found
  - x1 = 20, x2 = 20 with maximum profit at
    z = 4(20) + 3(20) = $140 per week

z	x1	x2	X3	X4	rhs	BV	ratio
1	0	0	2	1	140	z = 140	
0	0	1	2	-1	20	x_2 = 20	
0	1	0	-1	1	20	x_1 = 20	

Solving LP problems

- Simplex method
  – Easy to use but to solve large problems need to use computer
- Many software packages implement LP simplex method
  – General math/stats packages: Matlab, Excel, Mathematica etc.
  – Specialized optimization packages : LINDO, AMPL/CPLEX, XPRESS-LP, etc.
- Consider three examples
  1. AMPL/CPLEX : modeling language (and software) for designing and solving large complex LP/IP problems.
  2. MATLAB: General mathematics solver
  3. MS EXCEL with SOLVER (standard spreadsheet tool)

Example: Simplex Algorithm

- Look at the LP problem (slide 12,13) solved graphically:

Maximize: $25X_B + 30X_C$

Subject to: $(1/200)X_B + (1/140)X_C \le 40$

$$0 \le X_B \le 6000$$
$$0 \le X_C \le 4000$$

- Adding slack variables (S_1, S_2, S_3) and covert LP to standard form

Maximize: $Z = 25X_B + 30X_C$

Subject to: $(1/200)X_B + (1/140)X_C + S_1 = 40$

$$X_B + S_2 = 6000$$
$$X_C + S_3 = 4000$$
$$X_B, X_C, S_1, S_2, S_3 \ge 0$$

Simple AMPL Example

- Typing AMPL's description into a file – *prod0.mod*

```
var   XB;
var   XC;
maximize Profit: 25*XB + 30*XC;
subject to Time:          (1/200) * XB + (1/140) * XC  <=  40;
subject to B_limit:       0  <=  XB  <= 6000;
subject to C_limit:       0  <=  XC  <= 4000;
```

- Call AMPL commands:

```
ampl: model prod0.mod
ampl: solve;
MINOS 5.5: optimal solution found
2 iterations, objective 192000
ampl: display XB, XC;
XB  =  6000
XC  =  1400
ampl: quit
```


Matlab Example

- Look at the LP problem (slide 12,13) solved graphically:

Maximize: $25X_B + 30X_C$

Subject to: $(1/200)X_B + (1/140)X_C \le 40$

$0 \le X_B \le 6000$

$0 \le X_C \le 4000$

In Matlab formulate LP problems as

Minimize fx
s.t. Ax ≤ b

```
>> f = [-25 -30];
>>A = [1/200 1/140; 1 0; 0 1];
>> b = [40 6000 4000];
>> b = b';

>> [x,fval]=linprog(f,A,b)
Optimization terminated.
x =
 1.0e+003 *
    6.0000
    1.4000

fval = -1.9200e+005
```

Steps in Implementing an LP Model in Excel

1. Organize the data for the model on the spreadsheet.
2. Reserve separate cells in the spreadsheet to represent each decision variable in the model.
3. Create a formula in a cell in the spreadsheet that corresponds to the objective function.
4. For each constraint, create a formula in a separate cell in the spreadsheet that corresponds to the left-hand side (LHS) of the constraint.

- After entering LP model in spreadsheet, use **Solver** to solve the model

EXCEL Example

- Two types of leather belts: deluxe and regular, each requires 1 sq. yard of leather
- Each week: only 40 sq yard leather & 60 hrs of labor skill available
- Regular belt: 1 hr labor -> $3 profit
- Deluxe belt: 2 hr labor -> $4 profit
- Variable:
  - x1 = # deluxe belts produced/wk
  - x2 = # regular belts produced/wk
- LP: maximize profit z = 4x1 + 3x2

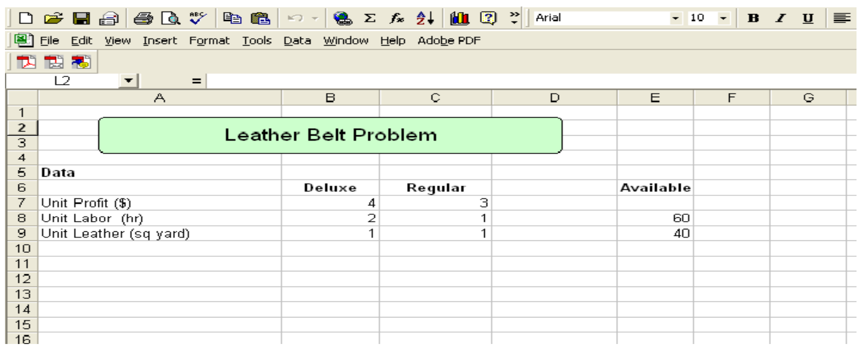
  s.t. 2x1 +x2 <= 60, labor constraint

  x1 + x2 <= 40, leather constraint

  x1, x2 >= 0, lower bound

Implementing LP model in a Spreadsheet

1. Organizing the data
   - Enter data into spreadsheet – in this example: unit profit, unit labor required, unit leather required and available resource
   - Coefficient (e.g., in the objective function and constraints) and values (e.g., RHS) of the model will be referred to or calculated from these data.
   - This makes the spreadsheet model more flexible to data changes.

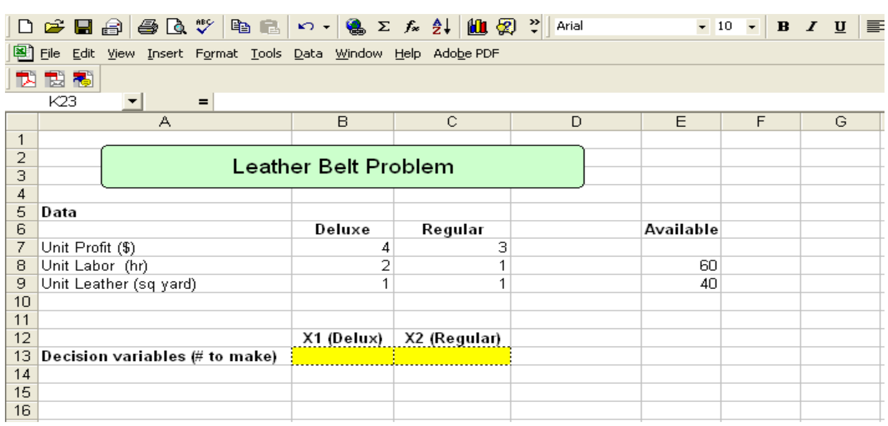


Leather Belt Problem

Row	A	B	C	D	E
5	Data				
6		Deluxe	Regular		Available
7	Unit Profit ($)	4	3		
8	Unit Labor (hr)	2	1		60
9	Unit Leather (sq yard)	1	1		40

LP in MS EXCEL

2. Representing the decision variables
   - In this example, let cells B13 and C13 represent the decision variables X1 and X2.
   - These cells should be shaded and outlined to visually distinguish them from other elements of the model.
   - Solver will determine the optimal values for these cells



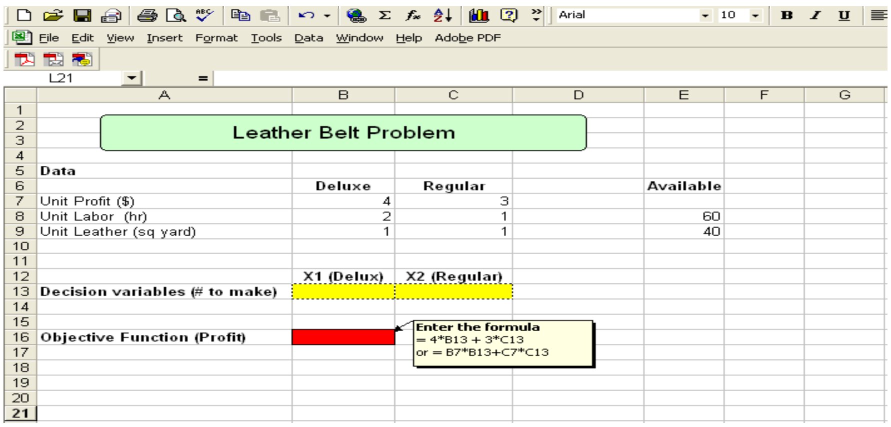
Leather Belt Problem

Row	A	B	C	D	E
5	Data				
6		Deluxe	Regular		Available
7	Unit Profit ($)	4	3		
8	Unit Labor (hr)	2	1		60
9	Unit Leather (sq yard)	1	1		40
12		X1 (Delux)	X2 (Regular)		
13	Decision variables (# to make)				

LP in MS EXCEL

3. Representing the objective function.

- Let Cell B16 represent an objective function: 4X1+3X2
  - Enter the formula in cell B16 as = B7*B13+C7*C13
  - Again, shaded and outlined to distinguish from other cells

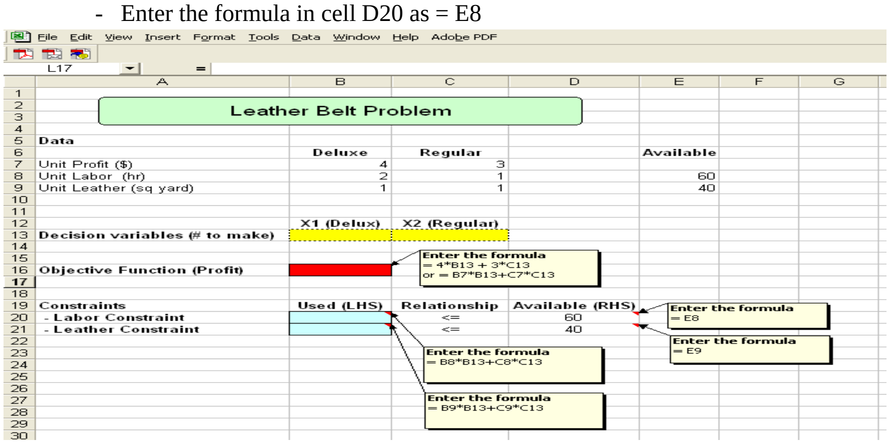


Telcom 2110 33

LP in MS EXCEL

4. Representing the Constraints (1)

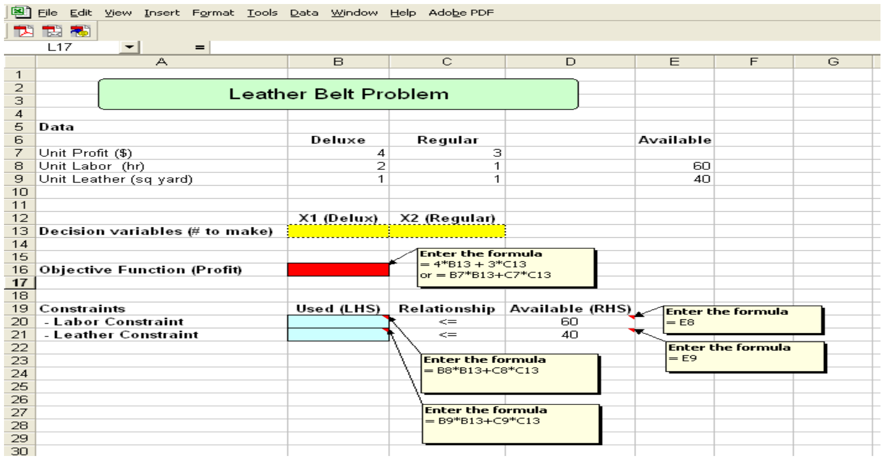
- Create a formula in a cell that corresponds to the LHS or RHS of the constraint
- Let cell B20 represent a LHS of labor constraint: 2X1+X
  - Enter the formula in cell B20 as = B8*B13+C8*C13
- Let cell D20 represent a RHS of labor constraint: 60
  - Enter the formula in cell D20 as = E8



Telcom 2110 34

LP in MS EXCEL

4. Representing the Constraints (2)
   - Let cell B21represent a LHS of leather constraint: X1+X2
     - Enter the formula in cell B21 as = B9*B13+C9*C13
   - Let cell D21 represent a RHS of leather constraint: 40
     - Enter the formula in cell D21 as = E9

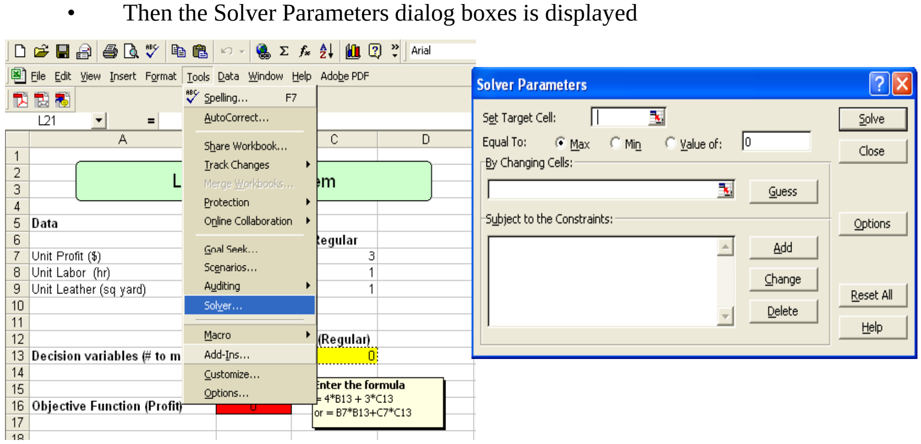


Telcom 2110 35

Using Solver

- After implementing an LP model in a spreadsheet, use Solver to solve the model.

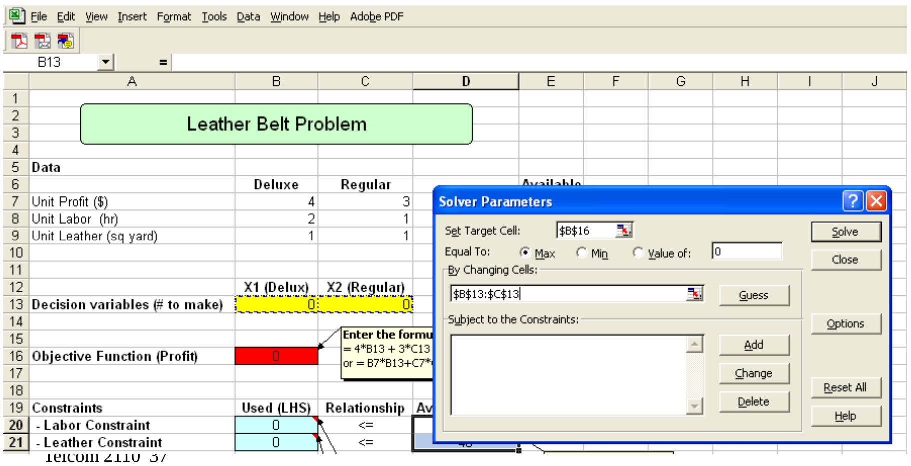
1. Invoking Solver in Excel
   - Choose the **Solver** command from the **Tools** menu
   - Then the Solver Parameters dialog boxes is displayed



Telcom 2110 36

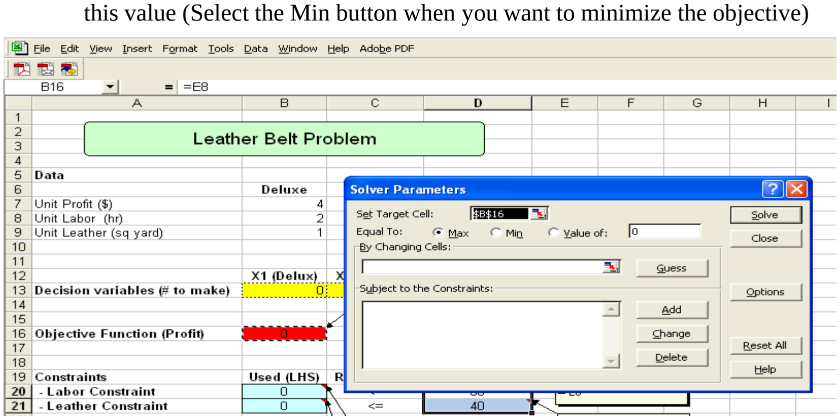
Using Solver

3. Defining the Changing Cells
   - Indicate which cells represent the decision variables in the model by entering their locations in the **By Changing Cells** box
   - In this example, cells B13 and C13 represent the decision variables



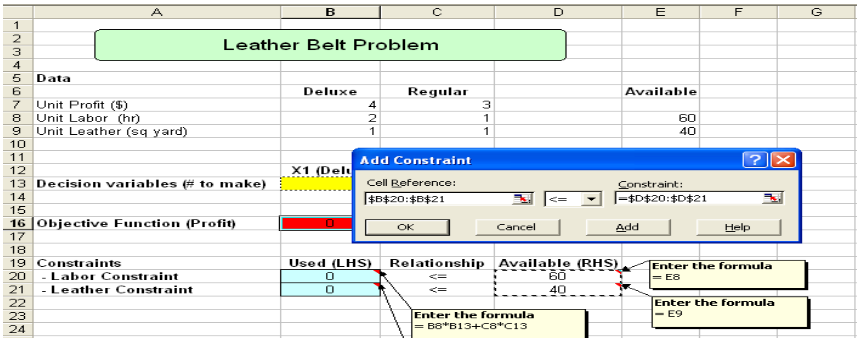
Using Solver

2. Defining the Target Cell
   - Specify the location of the cell that represents the objective function by entering it (B16 in this example) in the **Set Target Cell** box
   - Cell B16 contains a formula representing the objective function
   - Select the **Max** button, as in this example we want Solver to try to maximize this value (Select the Min button when you want to minimize the objective)



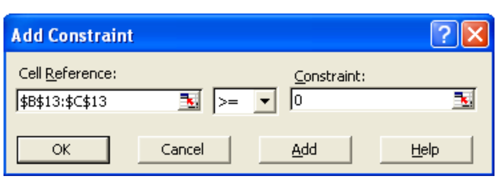
Using Solver

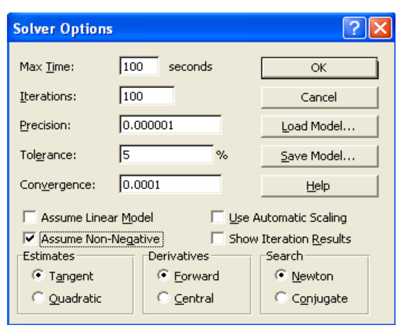
4. Defining the Constraint Cells
   - In the **Solver Parameters** box, click the **Add** button to define the constraint cells.
   - In the **Add Constraint** dialog box, specify the locations of cells that represent the LHS of a constraint in the **Cell Reference** box, and the RHS of a constraint in the **Constraint** box, and define the constraint type (<=, =, or >=). Click **Add** button again to define additional constraints.
   - In this example, cells B20 and B21 represent LHS cells whose values must be less than or equal to the values in cells D20 and D21(RHS cells) respectively.
   - Click the **OK** button when finished defining all constraints.



Using Solver

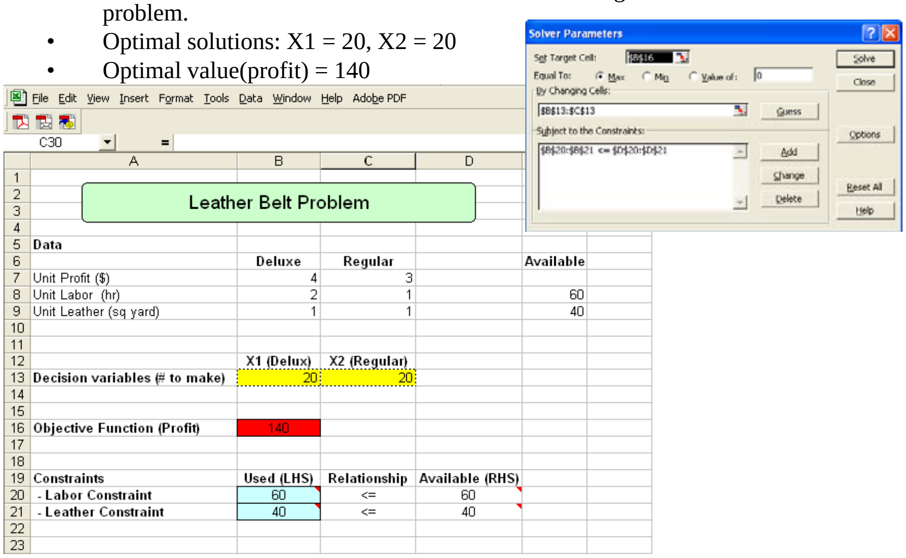
5. Defining the Nonnegativity Conditions/Constraints
   - The decision variables must be greater than or equal to zero
   - To add this constraint to the model,
     - specify locations of decision variable cells to the **Cell Reference** box (B13 through C13 in this example)
     - select a constraint type => from the dropdown box,
     - and enter a numerical constant “0” in the **Constraint** box
   - Alternatively, the nonnegativity can be imposed by checking the **Assume Non-Negative** check box in the **Solver Option** dialog box.





Using the Solver

6. Solving the model
   - Click the **Solve** button in the Solver Parameters dialog box to solve the problem.
   - Optimal solutions: X1 = 20, X2 = 20
   - Optimal value(profit) = 140



	A	B	C	D	
5	Data				
6		Deluxe	Regular		Available
7	Unit Profit ($)	4	3		
8	Unit Labor (hr)	2	1		60
9	Unit Leather (sq yard)	1	1		40
12		X1 (Delux)	X2 (Regular)		
13	Decision variables (# to make)	20	20		
16	Objective Function (Profit)	140			
19	Constraints	Used (LHS)	Relationship	Available (RHS)	
20	- Labor Constraint	60	<=	60	
21	- Leather Constraint	40	<=	40	

Integer Programming

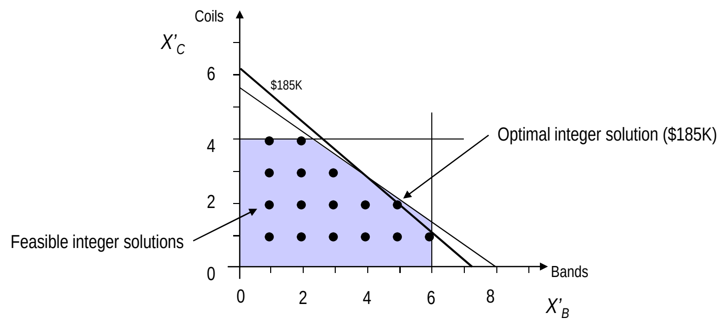
- Many problems in Network Design involve variables that are restricted to Integer Values – problems with such constraints are called Integer Programs or Mixed Integer Programs when some variables are integer and some are not
- Consider previous LP Example (slides 12-13)
  - Assume that orders for bands and coils are placed (and filled) in 1,000s of pounds only.
  - Although feasible region is greatly reduced, problem becomes much more difficult.
- New Symbolic Model
  - Let the new decision variables be the number of 1000 pound "units" or orders of bands and coils.

Maximize: $25000X'_B + 30000X'_C$

Subject to:
$$(1000/200)X'_B + (1000/140)X'_C \le 40$$
$$0 \le X'_B \le 6, \quad \text{integer}$$
$$0 \le X'_C \le 4, \quad \text{integer}$$

Integer Programming

- Graphical Solution Method

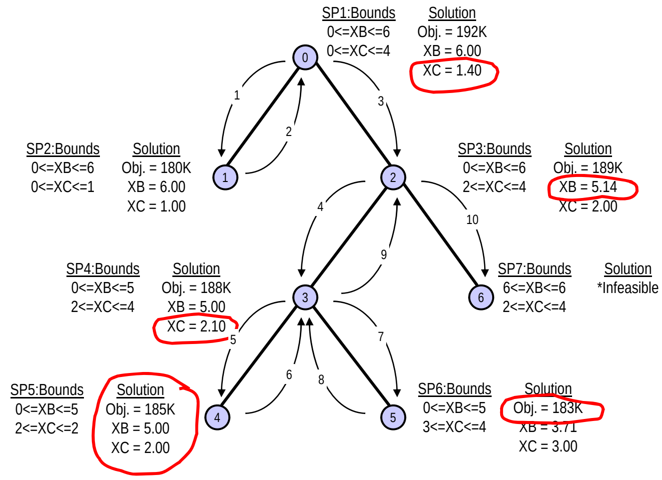


Solving IP Problems

- Branch-and-Bound Procedure
  - The solution space consists of a tree of LP subproblems, in which each integer variable is either fixed or its integrality constraint is “relaxed.”
  - The root node of the tree is the LP relaxation of the problem, i.e. all integer variables are relaxed.
  - The relaxation can result in an all integer solution, or a fractional solution (some decision variables are non-integer).
  - If the solution of the relaxation has fractional-valued integer variables, a fractional variable is selected for branching and two new subproblems are generated, each with more restrictive bounds on the branching variable.
  - The subproblems can result in an all integer solution, an infeasible problem or another fractional solution.
  - If the solution is fractional, the process is repeated.
  - Branches are “fathomed” if the subproblem is infeasible, the objective value is worse than the current best integer solution or the subproblem gives an integer solution.

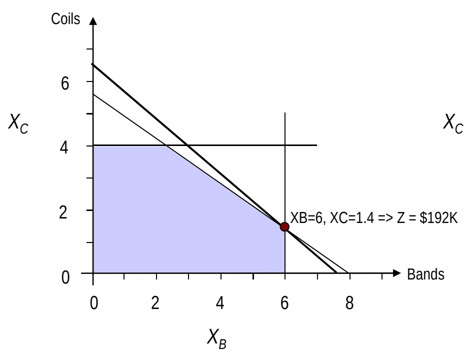
Solving MIP Problems

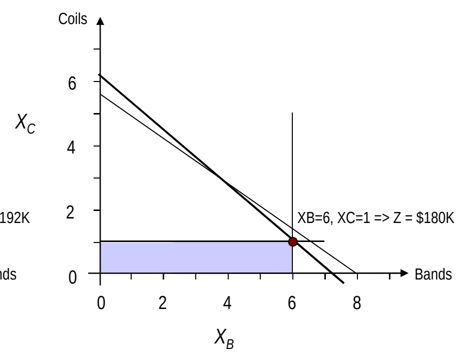
- Branch-and-Bound Tree (Example)



Branch-and-Bound (cont'd)

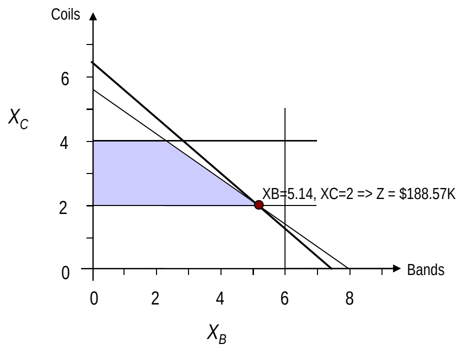
- Subproblem1: start with optimum LP solution
- Subproblem2: 0<=XB<=6
  0<=XC<=1



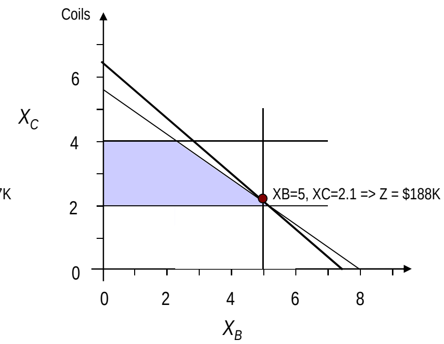


Branch-and-Bound (cont'd)

- Subproblem3: 0<=XB<=6
  2<=XC<=4



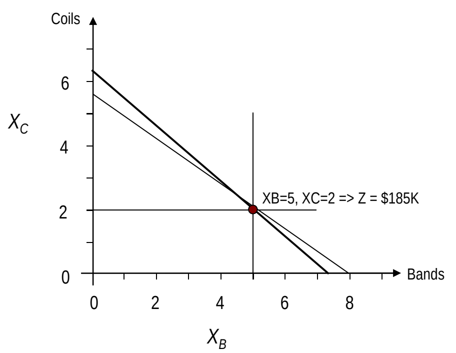
- Subproblem4: 0<=XB<=5
  2<=XC<=4



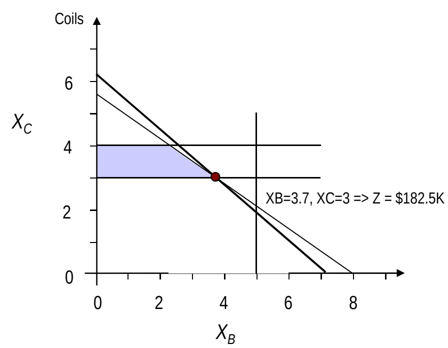
Telcom 2110 47

Branch-and-Bound (cont'd)

- Subproblem5: 0<=XB<=5
  2<=XC<=2



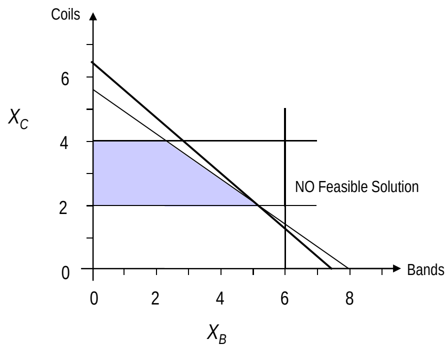
- Subproblem6: 0<=XB<=5
  3<=XC<=4



Telcom 2110 48

Branch-and-Bound (cont'd)

- Subproblem7: 6<=XB<=6
  2<=XC<=4



Matrix Expression of LP/IP Problems

- Why use it?
  - Most LP/IP problems are quite large and it becomes very cumbersome to describe them by explicitly giving each linear function, equality, and inequality in full.
  - It is desirable to model problems in a more general fashion (e.g. give an IP for optimally designing a mesh-restorable network in general as opposed to doing so for a specific network).
  - Formulate with Matrix Vector or Sum format

Algebraic Expression of LP/IP Problems

- Basic Production Model (Revisited)

Original Model	Algebraic Model
Problem name: prob.lp Maximize 25 XB + 30 XC Subject To 0.005 XB + 0.007143 XC <= 40 Bounds 0 <= XB <= 6000 0 <= XC <= 4000 End	Given: P, a set of products $a_j =$ tons per hour of product j, for each $j \in P$ $b =$ hours available at the mill $c_j =$ profit per ton of product j, for each $j \in P$ $u_j =$ maximum tons of product j, for each $j \in P$ Define variables: $X_j =$ tons of product j to be made, for each $j \in P$ Maximize: $\sum_{j \in P} c_j X_j$ Subject to: $\sum_{j \in P} (1/a_j) X_j \le b$ $0 \le X_j \le u_j, \quad \forall j \in P$

AMPL model

- Basic AMPL model (revisited) *-prod0.mod*

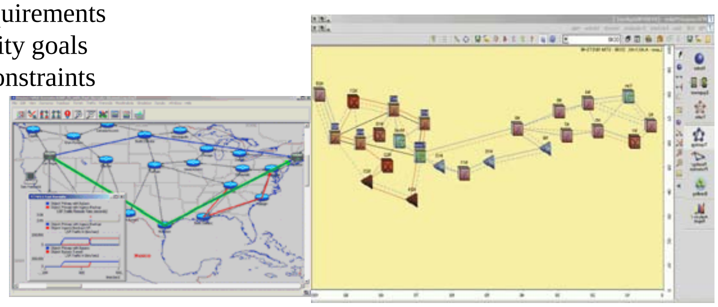
```
set P;
param a {j in P};
param b;
param c {j in P};
param u {j in P};
var X {j in P};
maximize Total_Profit: sum {j in P} c[j] * X[j];
subject to Time: sum {j in P} (1/a[j]) * X[j] <= b;
subject to Limit {j in P}: 0 <= X[j] <= u[j];
```

- model data *– prod0.dat*

```
set P :=  bands   coils;
param:        a       c       u  :=
bands        200      25    6000
coils        140      30    4000;
param  b := 40;
```

Network Design

- Many optimization formulations and design tools for network design (WANDL, VPISystems, Opnet)
  - Optimization Techniques usually form the initial basis of the formulation
  - Often use a heuristic or meta-heuristic solution techniques
- Formulation depends
  - Network layer (e.g., WDM, SONET, MPLS, etc.),
  - Technology (wired vs. wireless, etc.)
  - QoS requirements
  - Reliability goals
  - Other constraints



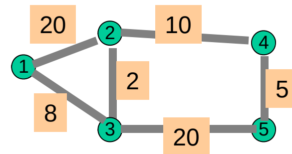
Network Design Optimization Models

- Network design problems
  - Network topology, Switch location, link capacity sizing, routings, etc..
- Good References
  - *M. Pioro and D. Medhi , Routing, Flow and Capacity Design in Communication and Computer Networks*
  - *ITU Network Planning Manual Pt1 and Pt 2*
- Consider a couple of simple examples here –more later
  - Max Flow Assignment (routing problem)
    - Arc formulation
    - Path formulation
  - Simple Network Design Model

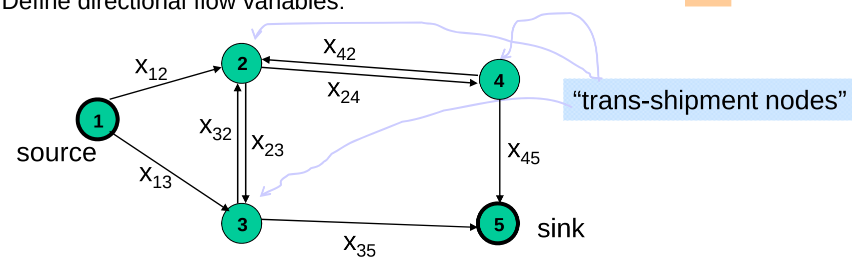
Max Flow LP Formulations

Example: LP to find max flow between nodes 1-5

Edge capacities:



Define directional flow variables:



Source: W. D. Grover, ECE 681, UofA, Fall 2004

Max Flow LP Formulations (2)

"transportation problem" or "arc-flow" approach

To maximize (1 → 5) flow : /* using lp_solve syntax */

max: x12 + x13 ; /* (or x35 + x45) */

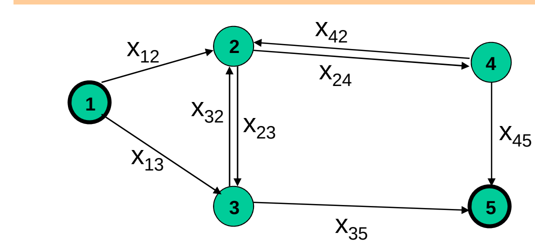
subject to constraints:

c1: x12 + x13 = x45 + x35 ; /* source = sink */

c2: x12 + x32 - x23 + x42 - x24 = 0 ; /* node 2 trans-shipment*/

c3: x13 + x23 - x32 - x35 = 0 ; /* node 3 trans-shipment*/

c4: x24 - x45 - x42 = 0 ; /* node 4 trans-shipment*/



Source: W. D. Grover, ECE 681, UofA, Fall 2004

Max Flow LP Formulations (3)

"transportation problem" or "arc-flow" approach

Continued....

Also subject to (capacity constraints):

x12 < 20 ;

x13 < 8 ;

x24 < 10 ;

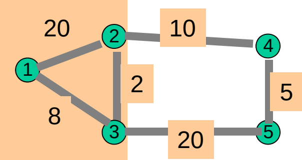
x42 < 10 ;

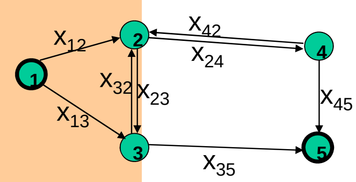
x23 < 2 ;

x32 < 2 ;

x35 < 20 ;

x45 < 5 ;





Source: W. D. Grover, ECE 681, UofA, Fall 2004

Max Flow LP Formulations (4)

"transportation problem" or "arc-flow" approach

Symbolically....

$$\max \sum_{1,i \in E} x_{1i}$$

s.t.

$$\sum_{1,i \in E} x_{1i} = \sum_{5,i \in E} x_{i5}$$

$$\sum_{i,j \in E} x_{ij} = \sum_{i,j \in E} x_{ji} \quad \forall i \in \{N - \{1,5\}\}$$

$$0 \le x_{ij} \le s_{ij} \quad \forall ij \in E$$

Where:

E = set of edges that exist

N = set of nodes

s_{ij} = spare capacity on edge *ij* (= s_{ji})

Source: W. D. Grover, ECE 681, UofA, Fall 2004

Max Flow LP Formulations

Alternate approach: "flow assignment to routes" or "arc-path" approach

Symbolically....

$$\max \sum_{i \in P_{15}} f_i$$

s.t.

$$\sum_{i \in P_{15}} f_i \cdot \delta_i^k \le s_k \qquad \forall k \in E$$

$$f_i \ge 0 \qquad \forall i \in P_{15}$$

Where:

E = set of edges that exist (indexed by k)

P_{15}= set of "eligible" distinct routes between nodes 1 and 5 (source-sink)

s_k = spare capacity on edge k

$\delta_i^k = 1$ if the i^{th} distinct route crosses span k. Zero otherwise

See Numerical Example posted on web pages

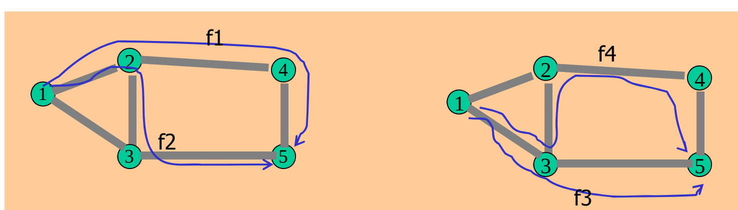
Source: W. D. Grover, ECE 681, UofA, Fall 2004

Telcom 2110 59

Network Flow LP Formulations (6)

"flow assignment to routes" or "arc-path" approach - example

Identify all distinct routes between source- sink (set P_{15}) ....



Route	associated flow variable
1-2-4-5	f1
1-2-3-5	f2

Route	associated flow variable
1-3-5	f3
1-3-2-4-5	f4

Source: W. D. Grover, ECE 681, UofA, Fall 2004

Telcom 2110 60

Network Flow LP Formulations (7)

"flow assignment to routes" or "arc-path" approach - example (2)

To maximize (1 -> 5) flow :

max: f1 + f2 + f3 + f4 ;

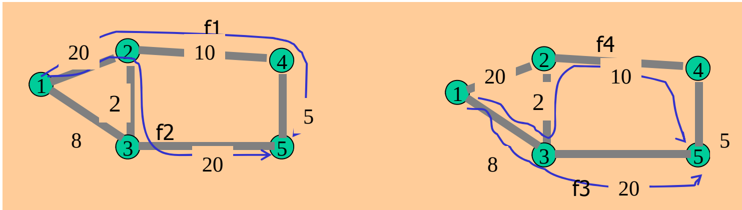
subject to constraints:

c1: f1 + f2 <= 20 ; /* link 12 capacity */

c2: f4 + f3 <= 8 ; /* link 13 capacity */

c3: f4 + f2 <= 2 ; /* link 23 capacity */

c4: f1 + f4 <= 10 ; /* link 24 */



Source: W. D. Grover, ECE 681, UofA, Fall 2004

Network Flow LP Formulations (8)

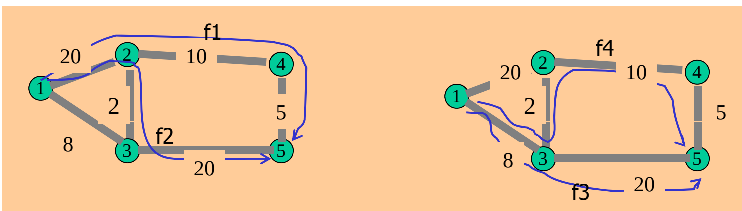
"flow assignment to routes" or "arc-path" approach - example (2)

What are the remaining constraints ? :

- for link 3-5 ... ?: c5: f3 + f2 <= 20; /*link 35 capacity */

- for link 4-5 ... ? c6: f4 + f1 <= 5; /*link 45 capacity */

(note this makes prior constraint f4 + f1<=10 redundant)



Source: W. D. Grover, ECE 681, UofA, Fall 2004

Network Flow LP Formulations (9)

"flow assignment to routes" or "arc-path" approach - example (3)

- Note that the δ_i^k "indicator" parameters do not appear explicitly in the executable model.
- Really they just represent our knowledge of the topology and the routes being considered.
- Implicitly above, we only wrote the flow variables that had non-zero coefficients.

Examples: $\delta_1^{12} = 1$ (flow1 crosses span 12) Hence f1 is in the first constraint

$\delta_3^{35} = 1$ (flow3 crosses span 35) Hence f3 is in the fifth constraint, etc.

See posted numerical example of arc-path approach

Source: W. D. Grover, ECE 681, UofA, Fall 2004

Simple Network Design Problem

- **indices**
  - $d=1,2,\ldots,D$ set of demands (source-destination pairs)
  - $p=1,2,\ldots,P_d$ possible paths for flows of demand d
  - $e=1,2,\ldots,E$ links
- **Input parameters (constants)**
  - h_d offered traffic load of demand d
  - c_e upper bound on capacity of link e
  - ξ_e unit (marginal) cost of link e
  - δ_{edp} = 1 if e belongs to path p realizing demand d; 0, otherwise
- **variables**
  - x_{dp} flow of demand d on path p
  - y_e capacity of each link

Capacitated flow allocation problem LP formulation – basic design

- **Objective:**

  minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

- **constraints**

  $\Sigma_p x_{dp} = h_d$ $\quad d=1,2,\ldots,D$

  $\Sigma_d \Sigma_p \delta_{edp} x_{dp} \leq y_e$ $\quad e=1,2,\ldots,E$

  $y_e \leq c_e$ $\quad e=1,2,\ldots,E$

  – flow and capacity variables are *continuous and non-negative*

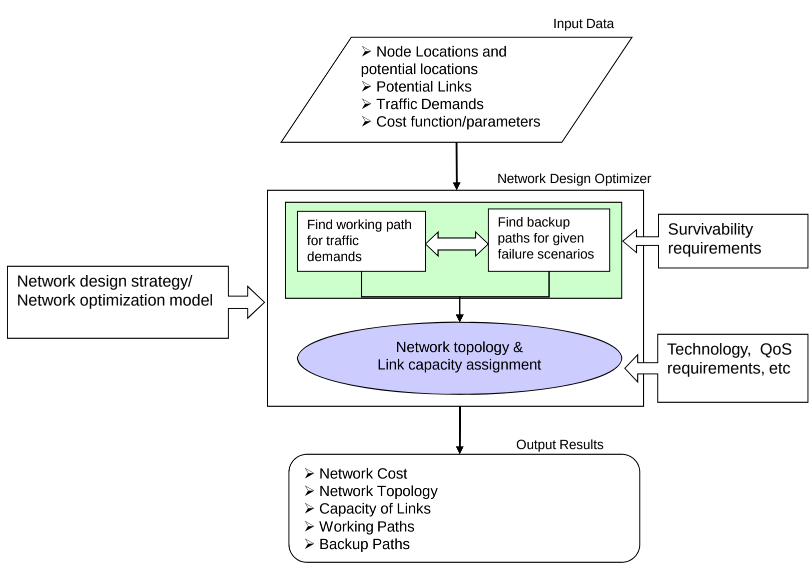
  $x_{dp} \geq 0, \quad y_e \geq 0$

This is an LP problem can solve using Simplex method

See posted numerical example.

Many variations tailor to specific network design problems → see posted slides from D. Medhi book

Optimization Based Network Design Procedure



Complexity

- Real Network Design problems are quite large (have many variables and constraints)
  - Graph Theory and Optimization Based algorithms for network design are complex – when can one use a technique?
- Complexity of an algorithm usually denotes O(.) which denotes the order of time growth in the algorithm as a function of problem variables
  - Dijkstra’s Algorithm for SPT O(N log(N)) where N is number of nodes in graph
  - Prim’s Algorithm for MST O(E log(N)) where N is # nodes, E # edges
- Problems that can be solved by a deterministic algorithm in a polynomial time complexity denoted P that is $O(N^k)$
- Problems that can not be solved with P complexity denoted NP and don’t scale well
  - Linear Programming Problems have P complexity
  - Integer Programming Problems have NP complexity
    - Still Branch and Bound can be used for small/medium problems !
- In general for NP problems use Sub-optimal solution algorithms (meta-heuristics – such as greedy alorithm, genetic algorithms, tabu search, etc.)

Summary

- Basic Constrained optimization
- Linear Programming
  - Formulation
  - Graphical Solution
  - Simplex Method
  - Software Tools
- Integer Linear Programming
  - Branch and Bound Solution
- Network Design Models
  - Arc Flow formulation
  - Path formulation
  - Many variations in the literature tailored to specific design problem