Final project: Integral Multi-Layer Networks design

Student: Tan Le
Course: EL 736
Term: Fall 2006
Outline

- Problem statement
- Modeling
- Solving by Ampl and Cplex
- Case study and analysis
- Proposed algorithms
- Conclusion
Problem statement

- Different technologies multi-layer networks
  - IP, ATM / MPLS / SONET
  - Circuit-switched voice / DCS (digital cross-connect systems) networks / SONET
  - Circuit-switched voice demand / traffic / Copper cable networks /
Problem statement

- Dimensioning at two Resource Layers

- Demand layer
  - demand carried by traffic network
  - continuous flows

- Traffic network layer
  - flow allocation to realize all demands
  - link capacity realized by transport layer
  - link capacities have modular values

- Transport network layer
  - flow allocation to realize all upper link capacity
  - link capacities have modular values

- Examples:
  - circuit-switched voice over DCS,
  - circuit-switched OC-3 services over WDM

**Figure 12.4** Two Resource Layer Network Example
Modeling of Two-Layer integral Dimensioning

**MIP: D/2L/MF(1)-CF(2)/BR/MC/LIN**

Two-Layer Dimensioning (Continuous/Integral Case)

**constants**
- $h_d$ volume of demand $d$
- $\delta_{edp} = 1$ if link $e$ of upper layer belongs to path $p$ realizing demand $d$; 0, otherwise
- $M$ size of the link capacity module in upper layer
- $\xi_e$ cost of one ($M$-module) capacity unit of link $e$ of upper layer
- $\gamma_{geq} = 1$ if link $g$ of lower layer belongs to path $q$ realizing link $e$ of upper layer; 0, otherwise
- $N$ size of link capacity module in lower layer
- $\kappa_g$ cost of one ($N$-module) capacity unit of link $g$ of lower layer

**variables**
- $x_{dp}$ (non-negative continuous) flow allocated to path $p$ realizing volume of demand $d$
- $y_e$ (non-negative integral) $M$-module capacity of upper layer link $e$
- $z_{eq}$ (non-negative integral) flow allocated to path $q$ realizing capacity of link $e$
- $u_g$ (non-negative integral) $N$-module capacity of lower layer link $g$

**objective**

minimize $F = \sum_e \xi_e y_e + \sum_g \kappa_g u_g$

**constraints**

\[
\begin{align*}
\sum_p x_{dp} &= h_d & d &= 1, 2, \ldots, D \\
\sum_q \sum_p \delta_{edp} x_{dp} &\leq M y_e & e &= 1, 2, \ldots, E \\
\sum_q z_{eq} &= y_e & e &= 1, 2, \ldots, E \\
M \sum_e \sum_q \gamma_{geq} z_{eq} &\leq N u_g & g &= 1, 2, \ldots, G
\end{align*}
\]
Solving by Ampl and Cplex

#Parameter indices

param D > 0 integer; #Demands
param pmax integer; #path numbers max for every demand in upper level
param E > 0 integer; #Links of upper level
param qmax integer; #path numbers max in lower level realizing every link e
param G > 0 integer; #links of lower level
param P{1..D} > 0 integer; #candidate paths in upper level for flow realizing demand d
param Q{1..E} > 0 integer; #candidate paths max in lower level for flow realizing link e

#Parameter constants

param h{1..D} >= 0; #volume of demand d
param delta {1..E,1..D,1..pmax} >= 0; #= 1 if link e of upper layer belongs to path p realizing demand d; 0, otherwise
param M >= 0; #size of the link capacity module in upper level
param epsilon{1..E} >= 0; #cost of 1 unit of link e in upper level
param gamma {1..G,1..E,1..qmax} >= 0; #= 1 if link g of lower layer belongs to path q realizing link e of upper layer; 0, otherwise
param N >= 0; #size of the link capacity module in lower level
param k{1..G} >= 0; #cost of 1 unit of link g in lower level

#Variables

var x{1..D,1..pmax} >= 0 integer; #flow allocated to path p realizing demand d
var y{1..E} >= 0 integer; #capacity of upper level link e
var z{1..E,1..qmax} >= 0 integer; #flow allocated to path q realizing capacity of link e
var u{1..G} >= 0 integer; #capacity of lower level link g

#Objective

minimize F: sum {e1 in 1..E} epsilon[e1]*y[e1] + sum {g1 in 1..G} k[g1]*u[g1];

#Constraints

subject to balance1 {d1 in 1..D}: sum {p1 in 1..P[d1]} x[d1 , p1] = h[d1];
subject to limit1 {e1 in 1..E}: sum {d1 in 1..D , p1 in 1..P[d1]} delta[e1 , d1 , p1] * x[d1 , p1] <= M * y[e1];
subject to balance2 {e1 in 1..E}: sum {q1 in 1..Q[e1]} z[e1 , q1] = y[e1];
subject to limit2 {g1 in 1..G}: sum {e1 in 1..E} M * (sum {q1 in 1..Q[e1]} gamma[g1 , e1 , q1] * z[e1 , q1]) <= N * u[g1];
Example

Input
- Directed Graph for all 3 layers
- D:=12;
- pmax:=6;
- E:=14;
- qmax:=7;
- G:=16;
- Topology on the beside figure.
- Size of link modul in upper level M=20;
- Size of link modul in lower level N=30;
- Demand

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>200</td>
<td>300</td>
<td>400</td>
<td>500</td>
<td>600</td>
<td>600</td>
<td>500</td>
<td>400</td>
<td>300</td>
<td>200</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

- Cost of 1 unit of link e in upper level

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Cost of 1 unit of link g in lower level

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>12</td>
<td>13</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Results

#### Output

F = 6725

<table>
<thead>
<tr>
<th>X</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>600</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>600</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Z</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>25</td>
<td>30</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>10</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>U</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>3</td>
<td>16</td>
<td>34</td>
<td>34</td>
<td>16</td>
<td>30</td>
<td>11</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td>14</td>
<td>11</td>
<td>20</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Case study

Input
- Directed Graph for all 3 layers
- $D := 28$;
- $p_{\text{max}} := 96$;
- $E := 48$;
- $q_{\text{max}} := 23$;
- $G := 60$;
- Topology on the beside figure.
- Size of link module in upper level $M = 10$;
- Size of link module in lower level $N = 10$;
- All demand equal to 100
- Cost of 1 unit of link $e$ in upper level all equal to 1
- Cost of 1 unit of link $g$ in lower level all equal to 1

Case study

- Output
  - $F = 3420$
  - Detail in .ans file
  - Example:
Case study

Demand layer

Demand numbering:
Demand 1: from node 1 to node 7
Demand 2: from node 1 to node 8
Demand 3: from node 2 to node 3
Demand 4: from node 2 to node 4
Demand 5: from node 2 to node 6
Demand 6: from node 2 to node 7
Demand 7: from node 2 to node 8
Demand 8: from node 2 to node 10
Demand 9: from node 3 to node 2
Demand 10: from node 3 to node 8
Demand 11: from node 4 to node 2
Demand 12: from node 4 to node 6
Demand 13: from node 4 to node 9
Demand 14: from node 5 to node 7
Demand 15: from node 5 to node 9
Demand 16: from node 6 to node 2
Demand 17: from node 6 to node 4
Demand 18: from node 6 to node 10
Demand 19: from node 7 to node 1
Demand 20: from node 7 to node 2
Demand 21: from node 7 to node 5
Demand 22: from node 8 to node 1
Demand 23: from node 8 to node 2
Demand 24: from node 8 to node 3
Demand 25: from node 9 to node 4
Demand 26: from node 9 to node 5
Demand 27: from node 10 to node 2
Demand 28: from node 10 to node 6

Upper link numbering:
Link 1: from node 1 to node 3
Link 2: from node 1 to node 8
Link 3: from node 4 to node 10

(R7) Demand layer
Case study

Upper layer

Demand layer

(R7)

(R10)
## Case study

### Upper layer

![Graph Image](image)

### Lower layer

![Graph Image](image)
Case study

Upper layer

Lower layer

(R10)

(A26)
Case study

Upper layer

Lower layer

(R10)

(A26)
Case study

Analysis

- Not optimal in separate layer but optimal in Global
- The optimal solution is based on the Shortest path allocation rule from the bottom layer to top layer
- From this problem, it could be expanded to N-layer linear optimization
- From this problem, it could be expanded to Multiple Modules multi-layer Dimensioning problem

Source: Morgan Kaufmann-Routing, Flow and Capacity Design in Communication and Computer Networks, Page 469
Problem of Two-Layer integral Dimensioning with multiple Modules

MIP: D/2L/MF(1)-CF(2)/BR/MC/MOD
(Link-Path) Formulation

Two-Layer Dimensioning Problem (Integral Case With Multiple Modules)

additional indices

\[ j = 1, 2, \ldots, J \] interface types at upper layer
\[ k = 1, 2, \ldots, K \] interface types at lower layer

constants

\[ h_d \] volume of demand \( d \)
\[ \delta_{edp} \] = 1 if link \( e \) of upper layer belongs to path \( p \) realizing demand \( d \); 0, otherwise
\[ M_j \] size of the link capacity module of type \( j \) in upper layer
\[ \xi_{ej} \] cost of one capacity unit of module type \( M_j \) of link \( e \) of upper layer
\[ \gamma_{geq} \] = 1 if link \( g \) of lower layer belongs to path \( q \) realizing link \( e \) of upper layer; 0, otherwise
\[ N_k \] size of link capacity module of type \( k \) in lower layer
\[ \kappa_{gk} \] cost of one capacity unit of module type \( N_k \) link \( g \) of lower layer

variables

\[ x_{dp} \] (non-negative continuous) flow allocated to path \( p \) realizing volume of demand \( d \)
\[ y_{ej} \] (non-negative integral) capacity units of upper link \( e \) for upper layer module type \( j \)
\[ z_{ejq} \] (non-negative integral) flow allocated to path \( q \) realizing capacity of type \( j \) of link \( e \)
\[ u_{gk} \] (non-negative integral) capacity of link \( g \) with lower layer module type \( k \)

objective

minimize \( F = \sum_j \xi_{ej} y_{ej} + \sum_k \kappa_{gk} u_{gk} \)

constraints

\[ \sum_p x_{dp} = h_d \] \( d = 1, 2, \ldots, D \)
\[ \sum_d \sum_p \delta_{edp} x_{dp} \leq \sum_j M_j y_{ej} \] \( e = 1, 2, \ldots, E \)
\[ \sum_q z_{ejq} = y_{ej} \] \( e = 1, 2, \ldots, E \)
\[ \sum_e \sum_j M_j \sum_q \gamma_{geq} z_{ejq} \leq \sum_k N_k u_{gk} \] \( g = 1, 2, \ldots, G \)
Solving by Ampl and Cplex

#Parameter indices

- param D>0 integer; #Demands
- param pmax integer; #path numbers max for every demand in upper level
- param E>0 integer; #Paths of upper level
- param qmax integer; #Path numbers max in lower level realizing every link e
- param G>0 integer; #Links of lower level
- param P{1..D} >0 integer; #Candidate paths in upper level for flow realizing demand d
- param Q{1..E} >0 integer; #Candidate paths max in lower level for flow realizing link e
- param J>0 integer; #Interface types at upper layer
- param K>0 integer; #Interface types at lower layer

#Parameter constants

- param h {1..D} >= 0; #Volume of demand d
- param delta {1..E,1..D,1..pmax}>= 0; #= 1 if link e of upper layer belongs to path p realizing demand d; 0, otherwise
- param M{1..J} >=0; #Size of the link capacity module of type j in upper layer
- param epsilon {1..E,1..J} >= 0; #Cost of one capacity unit of module type Mj of link e of upper layer
- param gamma {1..G,1..E,1..qmax}>= 0; #= 1 if link g of lower layer belongs to path q realizing link e of upper layer; 0, otherwise
- param N{1..K}>=0; #Size of link capacity module of type k in lower layer
- param kama {1..G,1..K} >= 0; #Cost of one capacity unit of module type Nk of link g of lower layer

#Variables

- var x {1..D,1..pmax} >=0 integer; #Non-negative continuous flow allocated to path p realizing volume of demand d
- var y {1..E,1..J} >=0 integer; #Non-negative integral capacity units of upper link e for upper layer module type j
- var z {1..E,1..J,1..qmax} >=0 integer; #Non-negative integral flow allocated to path q realizing capacity of type j of link e
- var u {1..G,1..K} >=0 integer; #Non-negative integral capacity of link g with lower layer module type k

#Objective

minimize F: sum {e1 in 1..E} (sum {j1 in 1..J} epsilon[e1,j1] * y[e1,j1]) + sum {g1 in 1..G} (sum {k1 in 1..K} kama[g1,k1]*u[g1,k1]);

#Constraints

subject to balance1 {d1 in 1..D}: sum {p1 in 1..P[d1]} x[d1,p1] = h[d1];
subject to limit1 {e1 in 1..E}: sum {d1 in 1..D} (sum{p1 in 1..P[d1]} delta[e1,d1,p1] * x[d1,p1]) <= sum {j1 in 1..J} M[j1] * y[e1,j1];
subject to balance2 {e1 in 1..E}: sum {q1 in 1..Q[e1]} z[e1,q1] = y[e1,j1];
subject to limit2 {g1 in 1..G}: sum {e1 in 1..E} (sum {j1 in 1..J} gamma[g1,e1,j1] * z[e1,j1,q1]) <= M[j1] * (sum {q1 in 1..Q[e1]} gamma[g1,e1,q1] * z[e1,j1,q1]) <= sum {k1 in 1..K} N[k1] * u[g1,k1];
Case study

Input

- Directed Graph for all 3 layers
- \( D = 28; \)
- \( p_{\text{max}} = 96; \)
- \( E = 48; \)
- \( q_{\text{max}} = 23; \)
- \( G = 60; \)
- Topology on the beside figure.
- Size of all link modules in upper level \( M_j = 10; \)
- Size of link modul in lower level \( N_k = 10; \)
- All demand equal to 100
- Cost of 1 capacity unit of all module type \( M_j \) of every link \( e \) of upper layer is 1
- Cost of 1 capacity unit of all module type \( N_k \) of every link \( g \) of lower layer is 1

Output

- \( F = 3420 \)
Proposed algorithms

- Flooding
  - Self-adaptive to topologies
  - Nodes are relatively independent
  - Have to prevent duplicate flooding

(Source: EL536 lecture node – Fall 2005)
Proposed algorithms

- Shortest path routing
  - Floyd’s Algorithm

```c
int floyds(int *matrix)
{
    int k, i, j;
    for (k = 1; k <= n; k++)
        for (i = 1; i <= n; i++)
            for (j = 1; j <= n; j++)
                if (Sh_pth[i][j] < (Sh_pth[i][k] + Sh_pth[k][j]))
                    Sh_pth[i][j] = Sh_pth[i][k] + Sh_pth[k][j];
}
```
Proposed algorithms

- Flow allocation
  - Bin packing problem
    - $N$ bins with capacity size $W_i$
    - $M$ things with size $s_j$ and value $v_j$
    - How to put things into bins to get maximum value

- Flow allocation
  - Demands $D$
  - $m$ Module with:
    - capacity size of module type $j$ : $M_j$
    - cost of one capacity unit of module $j$ : $C_j$
  - How to minimize cost to realize demand $D$
Proposed algorithms

- **Flow allocation**
  - **Greedy method**
    - Higher priority to pickup things that satisfy size constraint and have maximum value $R_j = v_j/s_j$
    - Quick but near optimum solution
  - **Bin packing algorithm**
    - Dynamic programming
      \[
      \text{Minvalue}(i,D) = \min(\text{Minvalue}(i-1,D - M[j]) + C[j]) \quad \text{for } j = 1..m
      \]
      
      Start with: \( \text{Minvalue}(i,D) = 0 \) if \( i=0 \) or \( D<=0 \)
    - However, DP algorithm **not work** if each module allowed to use no more than 1 time.
Conclusion

- Applying Bin-packing combine with Floyd’s Algorithm, we must get the same result as AMPL-CPLEX.
- If the cost model no more linearly depend on the flow capacity but depend on the distance, position allocation, cost of equipment... we could use other kind of methods like Heuristic or Convolution programming to solve./.