# CHAPTER 1 / NETWORK OPTIMIZATION

## 1.1 DEFINITIONS

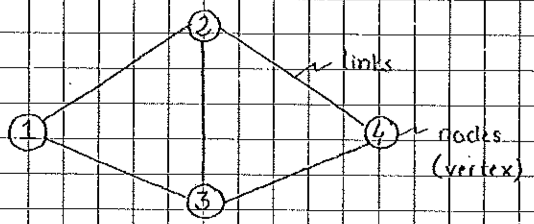
**Definition:** A network (graph) is an mathematical object, consisting of two components; first components; first component is the set of nodes (V) and a set of links (E,A)

**Notation:** $G = (V, E)$,

E
- Edges (direction is not imp.)
- Arcs (direction is important)

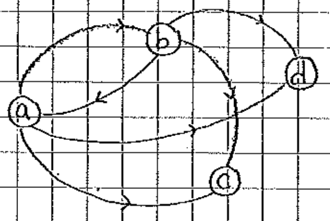
**EX:** Let $G = (V, E)$, $V = \{1, 2, 3, 4\}$ $E = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$ (be an undirected graph) $(1,2) = (2,1)$

Use this information to visualize this object.



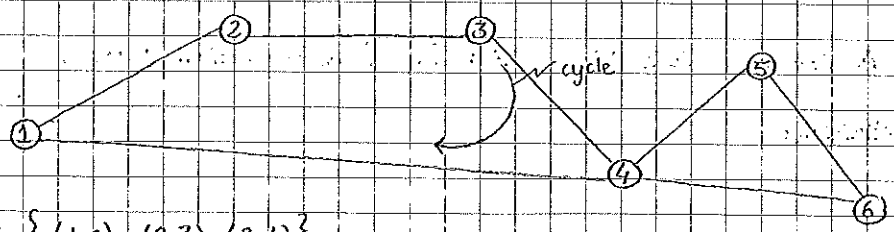

⇒ Visual representation of a network

**EX:** Let $G = (V, A)$, $V = \{a, b, c, d\}$ $A = \{(a,b), (b,c), (a,c), (b,a), (a,d), (b,d)\}$ (be a direct graph)



$P = \{(a,b), (b,c)\}$

**Definition:** A path $P$ in a graph $G = (V, E)$ is an ordered set of links $P \subseteq E$ such that the end point of link in $p$ is the starting point of $i$ link $i+1$ and no node is accounted twice.




$P_1 = \{(1,2), (2,3), (3,4)\}$

$P_2 = \{(1,2), (2,3), (3,4), (4,6)\}$

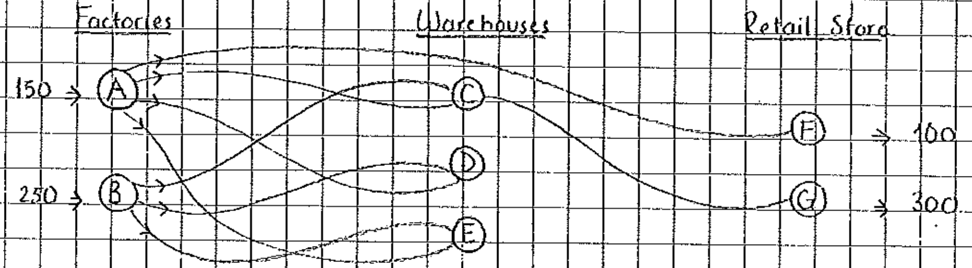
$\{(1,2), (2,3), (3,4), (4,1)\}$ is not a path

Q: What is a network for optimization Problem?

A: It is an optimization problem, that can be interpreted as a graph or network

EX: Furniture company X has two factories A and B, there where C and D, E houses and retail stores F and G.
On a given day, factory A supplies 150 units of furniture and 250 units of furniture.

Retail Stores: F demands 100 units of furniture and retail store G 300 units since its serving a large customer base. The factories can ship furnitures directly to the retail store, or ship to warehouses which will in turn ship to retail stores. It costs $C_{ij}$ to ship from location i to location j. What is the shipment plan for company X?




$G(V, A)$, $V(A, B, C, D, E, F, G)$

$$A = \{(A,C), (A,D), (A,E), (A,F), (A,G)$$
$$(B,C), (B,D), (B,E), (B,F), (B,G)$$
$$(C,F), (C,G), (D,F), (D,G), (E,F), (E,G)\}$$

Ingradients of a Network Optimization

1. Represent the network (graph) $G = (V, A)$ E
2. Define the decision variables
3. Define objective function
4. Write flow balance equation for each node

! If we use simplex, we'll got integer solutions. This is unique because generally simplex dont give us integers.

Network Optimization Problem ⇒ Integer solutions

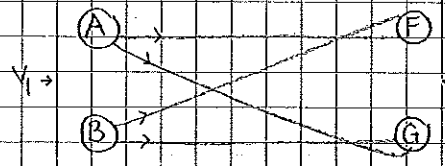
1.3 A SPECIAL CASE OF MCNF: The Transformation Problem

Definition: The transportation problem (TP) is a special case of (MCNF) where there is no transhipment made (supply, demand nodes only). In other words TP is defined over a graph.

$G = (V, A)$ where $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{\emptyset\}$ and

$\forall (i,j) \in A$, $i \in V_1$ and $j \in V_2$.

GIPTA




$V = \{(A,F), (A,G)$
$(B,F), (B,G)\}$

TP Min $\sum_{(i,j) \in A} C_{ij} X_{ij}$

St f.b equations for all supply and demand nodes $X_{ij} \geqslant 0$

! Property for MCNF (and also for TP) with integer supply and demand values, if you solve the problem using the simplex method of linear programming, you get integer optimal solutions (all decisions variables has int values) (Minimum cost network flow problem = MCNF)

Decision Variables: Amount of shipment from each source to each detemination.

$X_{ij}$ = Amount the total cost i to j.

EX $X_{AF}$

Objected: Minimize the total cost of shipping $Z = C_{AF} X_{AF} + C_{AE} X_{AG}$
$C_{AC} X_{AC} + C_{AD} X_{AD} + \ldots$

$$\text{Min} \sum_{(i,j) \in A} C_{ij} \overset{\geq 0}{X_{ij}} \qquad = \sum_{(i,j) \in A} C_{ij} X_{ij}$$

St $X_{ij} \geqslant 0 \quad (i,j) \in A$

A $150 = X_{AF} + X_{AC} + X_{AG} + X_{AD} + X_{AE}$
B $250 = X_{BF} + X_{BG} + X_{BC} + X_{BD} + X_{BE}$ Kirchoff
F $100 = X_{AF} + X_{BF} + X_{CF} + X_{DF} + X_{EF}$ Law
G $300 = X_{AG} + X_{BG} + X_{CG} + X_{DG} + X_{EG}$

↓ Flow Balance or flow conservation equations

What goes in what goes out

C $X_{AC} + X_{BC} = X_{CF} + X_{CG}$
D $X_{AD} + X_{BD} = X_{DF} + X_{DG}$
E $X_{AG} + X_{BE} = X_{EF} + X_{EG}$

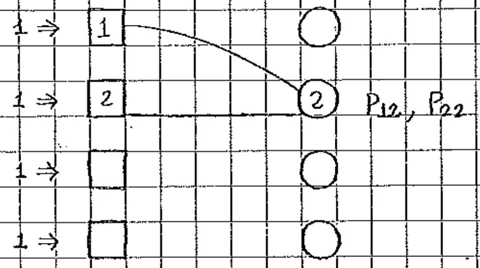
Feasibility issues for MCNF (for TP), For the problem to be feasible

Total Supply = Total Demand

GIPTA

## 1.3 THE ASSIGNMENT PROBLEM

EX: NASA has four astronauts to be employed at a space mission, for four posts on a space ship. The posts require diffrent qualifications (competencies)




The proficiency rating of ast. i arranged to post j is measured on $P_{ij}$.

Q: It is asked to find the optimal assignment of ast to post which gives the highest total prof rating while respecting the following requirements;

1. Each astronaut should be arranged to a single post.
2. Each post should have a single astronaut.

$$P = (P_{ij}) = \begin{array}{c|cccc} & ① & ② & ③ & ④ \\ \boxed{1} & 12 & 20 & 19 & 18 \\ \boxed{2} & 13 & 10 & 25 & 22 \\ \boxed{3} & 23 & 20 & 21 & 15 \\ \boxed{4} & 17 & 16 & 19 & 24 \end{array} \longrightarrow \text{proficiency rating of the astronaut.}$$

Decision Variable: $X_{ij} = \begin{cases} 1 & \text{if } i \text{ assigned to } j \\ 0 & \end{cases}$ → Binary Variable

Objective: $\text{Max} \sum_{i,j \in A} P_{ij} X_{ij}$

St.

(AP) $\sum_{j \in V_2} X_{ij} = 1 \quad \forall i \in V_1$

$\sum_{i \in V_1} X_{ij} = 1 \quad \forall j \in V_2$

$X_{ij} \geqslant 0$

$\text{Min} \sum_{(i,j) \in A} C_{ij} X_{ij}$

St. $\sum_{j \in V_2} X_{ij} = a_i \quad \forall i \in V$

$\sum_{i \in V_1} X_{ij} = b_j \quad \forall j \in V_2 \quad X_{ij} \geqslant 0 \quad \forall (i,j) \in A$

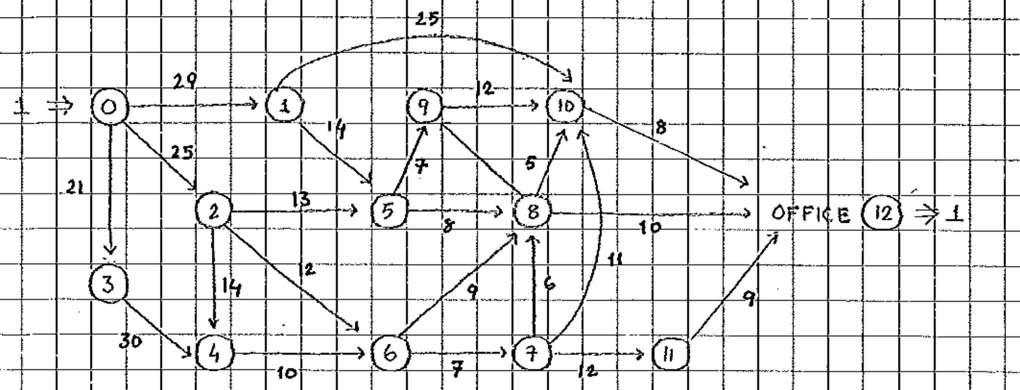
GIPTA

16.09.09
Wednesday

Observation: API and AP are identical equivalent because; (AP) is a special case of TP and TP is a special case of MCNF and we know that MCNF gives integer solutions when the data are integers

## 1.4 THE SHORTEST PATH PROBLEM

EX: Mr C. commutes to work every morning from home in the suburbs through a complicated road network, given below




For each road segment $(i,j)$, Mr. C estimates the time to cross this segment in the morning to be $t_{ij}$.
Find the path from ⓪ to ⑫ that will get Mr. C to minimum trip time.

Decision Variable: $x_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ is used in the trip by Mr. C} \\ 0 & \text{o.w} \end{cases}$ SPI

Objective Function: Total trip time for a path $= \sum_{(i,j) \in A} t_{ij} \, x_{ij}$ $\boxed{x_{ij} \geq 0}$ SP

S.t among all paths

(*) Constraints: $\sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = \begin{cases} 1 & i = 0 \\ 0 & \forall i \in \{1, \dots, 11\} \\ -1 & i = 12 \end{cases}$

$\delta^+(i) =$ set of all nodes that can be reached from $i$ by an arc $(i,j)$

$\delta^-(j) =$ set of all nodes $i$ that send an arc $(i,j)$ to $j$.

! Remark: Any 0-1 solution of equations (1) is a path from ⓪ to ⑫
Source node, sink node

Remark: SP and SPI are identical (equivalent), meaning if I solve SP I get an optimal solution that has only 0 or 1 values in the variables, because SP is a special case of MCNF with one supply node (the source) one demand node (the sink) and all nodes are transshipment nodes, and the data (RHS) are all integers.

GIPTA

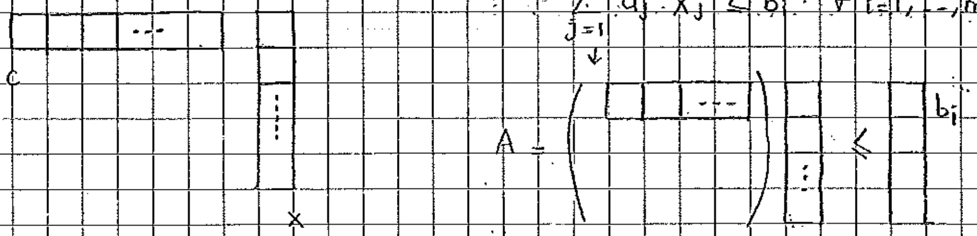
## 1.5 TOTALLY UNIMODULAR MATRICES OR.

Why don't we get solutions from MCNF with integer data?

Max $cx$ (inner product) $\Leftrightarrow$ Max $\sum_{j=1}^{n} c_j x_j$

St

$Ax \leq b$ $\qquad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \forall\, i=1,\dots,m$

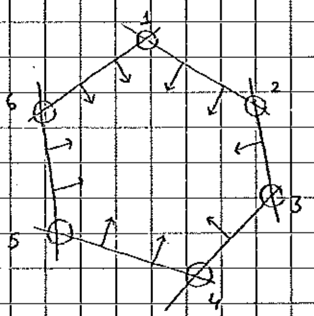



Back to IE202,

Remember that for a set $P = \{x : Ax \leq b\}$, and the associated linear programming problem;

$$\begin{array}{ll} \max & cx \\ \text{St.} & Ax \leq b \end{array} \qquad \begin{array}{l} \max\ cx \\ x \in P \end{array}$$

simplex method visits the extreme points of P and finds an optimal sol. that is an extreme point.




P is called a polyhedron.

Back to MCNF problem, the polyhdreon described by;

$$P_N = \left\{ \begin{array}{l} x_{ij} \geq 0, \\ \forall (i,j) \end{array} \ \sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = b_i \quad i=1,\dots,m \right\}$$

flow out − flow in = supply demand

**Theorem**: $P_N$ is an integer polyhdreon, i.e, it is a polyhdreon with all extreme points integer.

WHY?

**Definition**: A matrix $A \in \mathbb{R}^{m \times n}$ is called totally unnimodular (TU) if all sub determinants of A have value 0, 1, −1

**Theorem**: Let $A \in \mathbb{R}^{m \times n}$ be TU, and $b \in \mathbb{R}^m$ be an integer vector, then

$P = \{x : Ax \leq b\}$ is an integer polyhdreon.

! Theorem 2 does not answer the question WHY? Above, because $P_N$ has the form

$$P_N = \{Ax = b,\ x \geq 0\}$$

GIPTA

Remember: $\min \sum_{(i,j)\in A} c_{ij}x_{ij}$

$$\left.\begin{array}{l} \text{s.t.} \\ \displaystyle\sum_{j\in\delta^+} x_{ij} - \sum_{i\in\delta^-} x_{ki} = b_i \quad \begin{array}{l}+\text{supply}\\ -\text{demand}\end{array} \\ 0 \le x_{ij} \quad \forall (i,j)\in A \end{array}\right] \text{MCNF}$$

Fact 1: $A$ TU $\Leftrightarrow$ $G = \{x : Ax \le b\}$ ($b$ integer) is an integral polyhedron

Fact 2: If $A$ is TU, then $[A\ I]$ is TU

$A$ is TU, then $\begin{pmatrix} A \\ -A \end{pmatrix}$ is TU

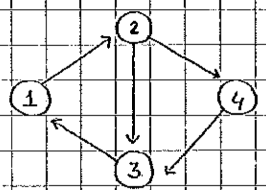
$A$ is TU, then $\begin{pmatrix} A \\ -A \\ I \\ -I \end{pmatrix}$ is TU

$$\left.\begin{array}{l} Ax \le b \\ -Ax \le b \end{array}\right) Ax = b$$
$$-x \le 0 \rightarrow x \ge 0$$

Application of Fact 1 to $\begin{pmatrix} A \\ -A \\ I \\ -I \end{pmatrix}$ is TU $P_2 = \left\{ x : \begin{pmatrix} A \\ -A \\ -I \end{pmatrix} x \le \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix} \right\}$

Fact 3: (follows from $F_1$ and $F_2$) If $A$ is TU then $P_2 = \{x : x \ge 0\ Ax = b\}$ is an integral polyhedron (for $b$ integer)

Structure of the A Matrix for MCNF

$b_1 \Rightarrow$  $\Rightarrow b_2$



$$(A)\ \left\{\begin{array}{lr} x_{12} + x_{13} = b_1 & (1) \\ -x_{12} + x_{23} + x_{24} = 0 & (2) \\ -x_{13} - x_{23} + x_{34} = 0 & (3) \\ -x_{24} - x_{34} = -b_4 & (4) \end{array}\right.$$

$b_1 > 0,\ b_4 > 0$

$$A = \begin{array}{c|ccccc} & x_{12} & x_{13} & x_{23} & x_{24} & x_{34} \\ \hline 1 & 1 & 1 & & & \\ 2 & -1 & & 1 & 1 & \\ 3 & & -1 & -1 & & 1 \\ 4 & & & & -1 & -1 \end{array} \begin{pmatrix} x_{12} \\ x_{13} \\ x_{23} \\ x_{24} \\ x_{34} \end{pmatrix}$$

$\rightarrow$ node arc incidence matrix of a graph $G$ is a network

- Each column has two nonzero entries and then two entries are +1 and −1.

Fact 4: Node arc incidence matrix is TU

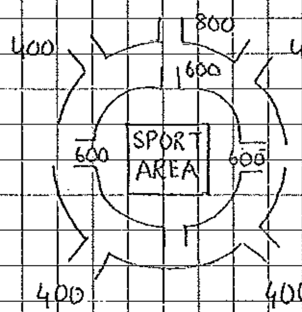
Fact 5: A matrix $A \in \{0, \mp 1\}$ is TU iff every collection of column of A can be partitioned into two sets such the sum of the columns in one part the sum of the columns in the other part is a vector with entries equal to 0, 1 or -1.

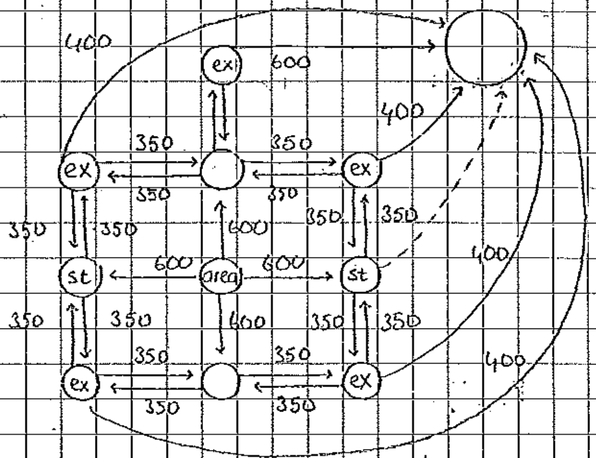
Fact 6: A TU $\Leftrightarrow$ $P_B = \{ l \leq x \leq u \mid Ax = b \}$ is integer poly because A is TU and

$$\begin{pmatrix} A \\ -A \\ I \\ -I \end{pmatrix} \text{ is also TU.}$$

## 1.6 MAXIMAL FLOWS IN NETWORKS

Evacuation Problem




- - - : imaginary
ex : exits
st : stairs

Find the amount of the flow from the area to safety

Formulation as an LP:

DV: Let $X_{ij}$ be the amount of the flow from node i to node j
$\vartheta$: be the total amount of flow into the sink node

Objective Function:

max $\vartheta$
St

$$\sum_{j \in \delta^+(i)} X_{ij} - \sum_{k \in \delta^-(i)} X_{ik} = 0 \quad \text{for all nodes except nodes Arena safety.}$$

$$\sum X_{0j} - \vartheta = 0 \qquad \vartheta \geq 0$$

$$\vartheta - \sum_{k \in \delta^-(s)} X_{kj} = 0 \qquad V_{ij} \geq X_{ij} \geq 0 \qquad V_{ij} : \text{capacity}$$

★ Therefore, with the reinterpretation of $\vartheta$ a flow variable of an imagined arc from safety (S) to arena (O), the problem is in the form MCNF, but with bounds on the flow variables. $X_{ij}$ no upper bound in $\vartheta$

Remark: It is easy to compute upper bounds on the maximal flow, $V^*$ i.e. I can find a number $k$ such that $V^* \leq k$.

Definition 1: A cut partition in $G=(V,A)$ is a partition of $V$ into two subsets with empty intersection ($S \cap T = \{\emptyset\}$) and $S \cup T = V$, $s \in S$ ↓ nodes to source
$t \in T$ → nodes to sink

Definition 2: A corresponding a cut partition $(S,T)$ is a subset of arcs $A$ $C = \{(i,j) \in A\}$, $i \in S$, $j \in T\}$

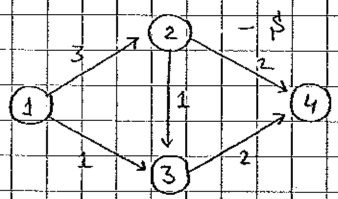
Definition 3: The capacity of a cut $Cap(C) = \sum_{(i,j) \in C} c_{ij}$

imp
1. for any cut $C$, $v^* \leq cap(C)$
2. there exits a cut $C^*$ s.t. $v^* = cap(C^*)$

EX: let $s = \{s, 1, 2, 3, 4, 5, j\}$ $C_1 = \{(1,6), (2,t), (1,7), (3,9), (5,8), (4,t)\}$
$t = \{6, 7, 8, 9, t\}$
$350 + 400 + 350 + 350 + 600 + 400 = 2450$

from here $v^* \leq 2450$
i.e $S = \{j\}$ $T = V \setminus S$ then $v^* \leq 2400$ so $v^* \leq 2400$

Maximal Flow Cont'd




very important Duality result

max flow = min cut

Easy to show $\boxed{\text{max flow} \leq \text{min cut}}$

Recall the formulation of Max-flow
max $v$
St.
$$\sum_{k \in \delta^+(i)} X_{ik} - \sum_{j \in \delta^-(i)} X_{ji} = \begin{cases} v & \text{if } i = \text{source} \\ 0 & \text{o.w} \\ -v & \text{if } i = \text{sink} \end{cases} \qquad (*)$$

Take any feasible flow $(\tilde{x}, \tilde{v})$ in the network.
Take a cut partition $(S,T)$ in the network.
Among the equations $(*)$ pick the ones corresponding to $S$.

Add these equations up.

$$\sum_{\substack{i \in S \\ j \in T}} \tilde{X}_{ij} - \sum_{\substack{i \in T \\ j \in S}} \tilde{X}_{ij} = \tilde{v} \Rightarrow \tilde{v} \leq \sum_{\substack{i \in S \\ j \in T}} U_{ij} \qquad v^* \leq \sum_{\substack{i \in S^* \\ j \in T^*}} U_{ij}$$

$$
\begin{aligned}
X_{12} + X_{13} &= v \\
-X_{12} + X_{23} + X_{24} &= 0 \\
\hline
X_{13} + X_{23} + X_{24} &= v \qquad \underline{v \leq 4}
\end{aligned}
$$

$$
\begin{aligned}
+X_{12} + X_{13} &= v \\
-X_{23} - X_{13} + X_{24} &= 0 \\
\hline
X_{12} + X_{34} - X_{23} &= v \qquad v \leq 5
\end{aligned}
$$

## 1.7 ALGORITHM FOR SHORTEST PATH

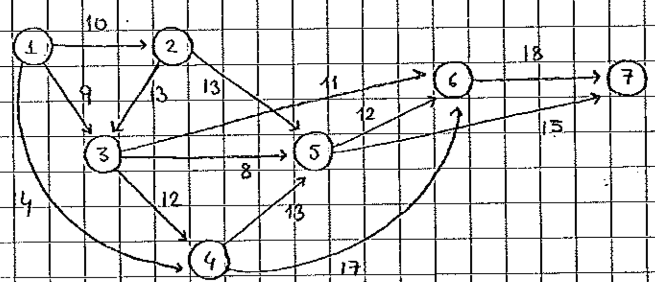
Dijkstras's algorithm

**Goal**: find the shortest path from a source node to all the other nodes in the graph

**Assumption**: All arc lenght area nonnegative.

**Idea**: Keep labels on all nodes, start with temporary labels on all nodes except the source, then at each step you chose another node and give it a permanent label and update the remaining temporary labels, continue untill all nodes receive a permanent label

EX:




Step 0

$u_1 = 0$ (Permanent) $\quad n = \#$ of nodes $\quad (n-2)+(n-3)+\dots+\frac{(n-1)(n-2)}{2}$

comparisons $\longrightarrow$

$u_j = a_{1j} \quad \forall j \setminus V\{1\}$

$$a_{ij} = \begin{cases} \text{lenght of arc } (i,j) & n^2 \\ +\infty & \text{if } (i,j) \notin A \end{cases}$$

$P = \{1\} \qquad T = \{2, 3, \dots, 7\}$

Step 1 (Choice of the next permanent label)

$u_2$: 10 $\quad$ Let $k$ be the index of the node with the smallest label

$u_3$: 9 $\quad P = P \cup \{K\} \quad T = T \setminus \{K\} \quad$ if $T = \emptyset$ stop

$u_4$: 14

$u_5$: $\infty$

$u_6$: $\infty$

$u_7$: $\infty \Rightarrow P = \{1, 3\}, \quad T = \{2, 4, 5, 6, 7\}$

Step 2 (Update the temporary labels)

$u_j = \min \{u_j, u_k + a_{kj}\} \ \forall j \in T$, go to step 1

n-2 addition, n-2 comparison $\Rightarrow 2(n-2) + 2(n-3) + 2(n-4) + \dots + 0 = \frac{2(n-1)(n-2)}{2}$

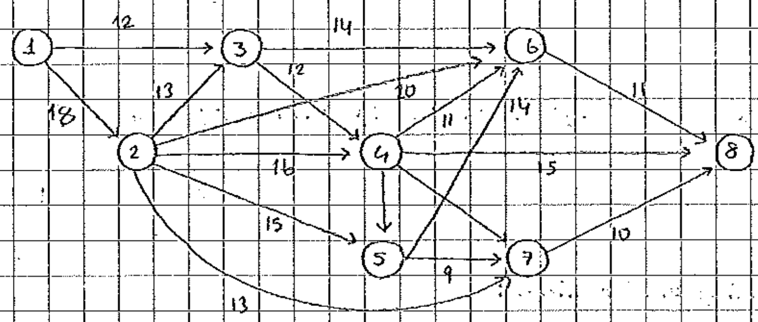
$u_2 : \min \{10, u_3 + a_{32}\} = 10$
$u_4 : \min \{14, u_3 + a_{34}\} = 14$
$u_5 : \min \{\infty, 9+8\} = 17$ $\Rightarrow$ new labels
$u_6 : \min \{\infty, 9+11\} = 20$ so,
$u_7 : \min \{\infty, 9+\infty\} = \infty$ $P = \{1,3,2\}, \ T = \{4,5,6,7\}$

$u_4 : \min \{14, u_2 + a_{24}\} = 14$
$u_5 : \min \{17, u_2 + a_{25}\} = 17$
$u_6 : \min \{20, u_2 + a_{26}\} = 20 \Rightarrow P = \{1,3,2,4,5\}, \ T = \{6,7\}$
$u_7 : \min \{\infty, \infty\} = \infty$

$\Downarrow$

$u_6 : \min \{20, 17+12\} = 20$
$u_7 : \min \{\infty, 17+15\} = 32 \Rightarrow P = \{1,3,2,4,5,6,7\}$

EX:




Step 0
$u_1 : 0$
$u_2 : 18$
$u_3 : 12$
$u_4 : \infty$
$u_5 : \infty$
$u_6 : \infty$
$u_7 : \infty$
$u_8 : \infty$

$P = \{1,3\} \quad T = \{2,4,5,6,7,8\}$

- $u_2 = 18, \ u_4 = 24, \ u_5 = \infty, \ u_6 = 26, \ u_7 = \infty, \ u_8 = \infty$

$P = \{1,3,2\} \quad T = \{2,4,5,6,7,8\}$

- $u_4 = 24 \quad u_5 = 33 \quad u_6 = 26 \quad u_7 = 31 \quad u_8 = \infty$

$P = \{1,3,2,4,6\} \quad T = \{5,7,8\}$

- $u_5 = 33 \quad u_8 = 37$

$P = \{1,3,2,4,6,7,5\}$

- $u_8 = 37$

$P = \{1,3,2,4,6,7,5,6\}$

## Complexity of an Algorithm

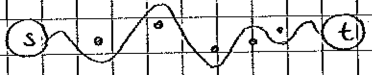
1. Code the size of the problem instance
2. Compute the number of operations ($+$, comparisons) thats required as a fcn of the problem size until the algorithm stops.

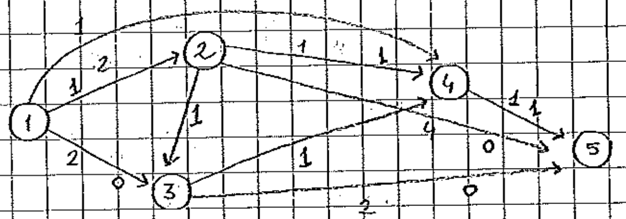
Comp of Dijkstra algorithm $= n^2 + \frac{(n-1)(n-2)}{2} + \frac{2(n-1)(n-2)}{2} = kn^2 + \dots + O(n^2)$

## 1.8 THE FLOW AUGMENTATION ALGORITHM FOR MAXIMUM FLOW

Idea: Start with an initial feasible (integer flow) and check whether this flow is the maximal flow. (•)

(•) Can you find a path from the source node to sink node on which the flow can be increased? If such a path exists, then its called a flow augmenting path, the current flow is not optimal.






left
we use

### Ford Fulkeson Flow Augmentation Algorithm for Max Flow

Step 0: Start with an integer feasible flow. Give labels $(-, \infty)$ to the source node $s$.

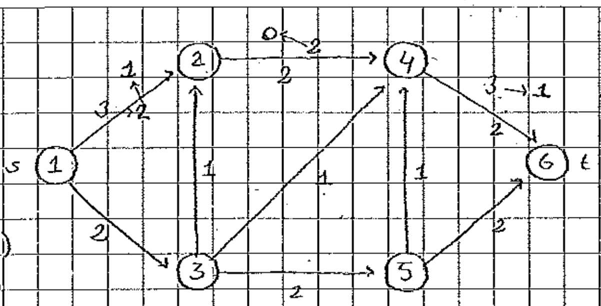
Step 1 (Scanning and Labelling):

1.1 If all labelled nodes have been scanned go to step 3.
1.2 For all labelled but unscanned nodes, scan as follows
For all arcs $(i,j)$, if $j$ is unlabelled and $X_{ij} < C_{ij}$ then give $j$ the label $(i^+, l_j)$ where $l_j = \min(l_i, C_{ij} - X_{ij})$
For all arcs $(j,i)$, if $j$ is unlabelled and $X_{ji} > 0$, then give $j$ the label $(i^-, l_j)$ where $l_j = \min(l_i, X_{ji})$

1.3 If node $t$ (sink node) has been labelled, then go to Step 2.

Step 3: A maximal flow has been found, (the current flow is maximum), the nodes that have a label constitute a set $S$, and the nodes without a label make up the set $T$ and $(S,T)$ form a maximal set.

EX:



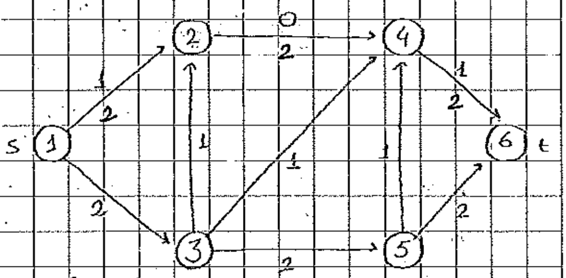

(−, ∞)
v = 2

- Start with the zero flow,

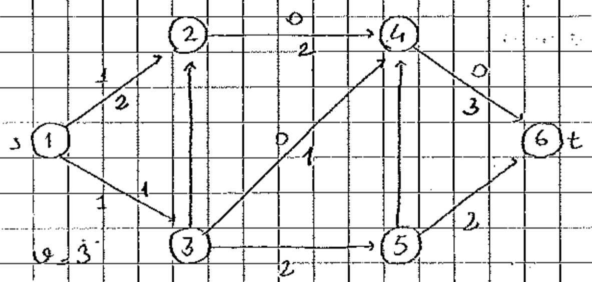
| | | | | |
|---|---|---|---|---|
| (1,2) | 2 ← | ($1^+$, min(∞, 3)) ← | ($1^+$, 3) | Forward arcs, minus signs |
| (1,3) | 3 ← | ($1^+$, 2) | | |
| (2,4) | 4 ← | ($2^+$, 2) | | |
| (3,5) | 5 ← | ($3^+$, 2) | | |
| (4,6) | 6 ← | ($4^+$, 2) | | |

- F.A.P has been found. $6 \rightarrow 4 \rightarrow 2 \rightarrow 1$ (+2, +2, +2)

Scan 1 ✓ (1,2) 2 ← ($1^+$, 1)
(1,3) 3 ← ($1^+$, 2)
Scan 2 ✓
Scan 3 4 ← ($3^+$, 1)
5 ← ($3^+$, 2)
Scan 4 6 ← ($4^+$, 1)



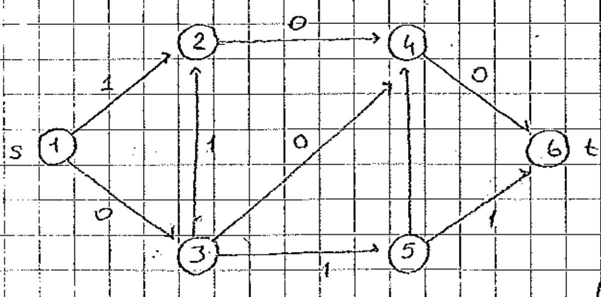

- Another F.A.P has been found $6 \rightarrow 4 \rightarrow 3 \rightarrow 1$ (+1, +1, +1)




v = 3

Scan 1 ✓ (1,2) 2 ← ($1^+$, 1)
(1,3) 3 ← ($1^+$, 1)
Scan 2 ✓ 5 ← ($3^+$, 1)
Scan 3 4 ← ($5^+$, 1)
Scan 5 6 ← ($5^+$, 1)

- Another F.A.P found $6 \rightarrow 5 \rightarrow 3 \rightarrow 1$ (+1, +1, +1)



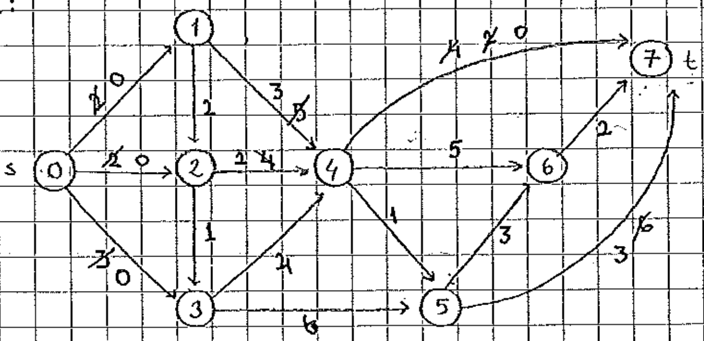

v = 4

Scan 1 ✓ (1,2) 2 ← ($1^+$, 1)
Scan 2 ✓

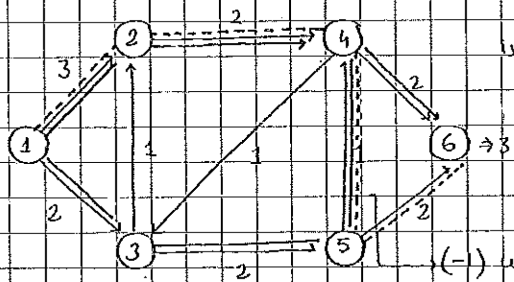
The current flow is optimal
(S,T) S = {1,2} T = {3,4,5,6}

We cannot scan 3 because it is labelled.

EX:



EX: Another example for max-flow
Given $G=(V,A)$ with arc capacities.



(−) we have to suppress

And the following feasible flow is given.

$x_{12}=1$ $x_{24}=1$ $x_{46}=2$
$x_{13}=2$ $x_{35}=2$ $x_{54}=1$
$x_{56}=1$

$v=3$

- Check whether the current flow is maximum flow from ① to ⑥

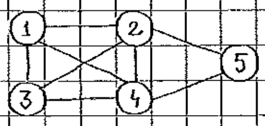
| | | | | | |
|---|---|---|---|---|---|
| +1 ① | $(-,\infty)$ | $x_{12}=2$ | $x_{13}=2$ | $x_{56}=2$ | |
| ② | $(1^+,2)$ | $x_{24}=2$ | $x_{35}=2$ | $x_{54}=0$ | |
| +1 ④ | $(2^+,1)$ | $x_{46}=2$ | | | |
| −1 ⑤ | $(4^-,1)$ | | $v=4$ | - - - - -: flow that we found. | |
| +1 ⑥ | $(5^+,1)$ | (Not optimal) | | ———: | |

- Feasible flow = node conservation, nonnegativity

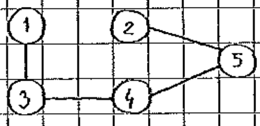
## 1.9 OPTIMUM TREES

Definition 1: Given a graph $G=(V,E)$, a forest is a spanning subgraph with no cycles.

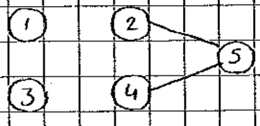
$G=(V,E) \rightarrow G'=(V,E')$



1)



2)



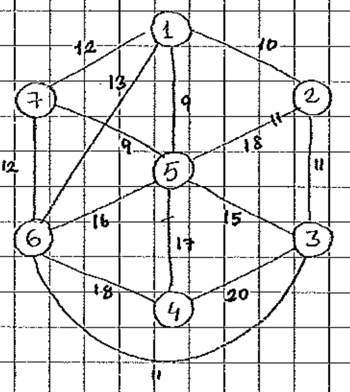
Definition 2: A cycle C is a graph $G=(V,E)$ is a sequence of edges starting out a given node and ending at the source node without visiting any other (arc, no edge is repeated) node twice.

Definition 3: A tree $T=(V,E')$ is a graph $G=(V,E)$ is a forest which is connected.
↓ there is a path to any node to every other node in T.

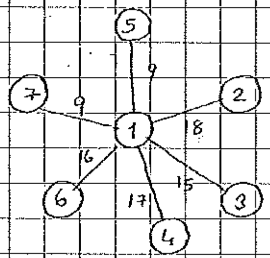
05.10.2009

**Theorem:** Let $G=(V,E)$ be an undirected and connected graph with $|V|=n$. A spanning subgraph $T=(V,E')$ with $E' \subseteq E$ is the tree iff
(a) it is connected
iff
(b) it has $(n-1)$ edges
iff
(c) there is a "unique" path in T from any node to every other node
iff
(d) adding any edge from $E \setminus E'$ results in the creation of a unique/single cycle in the new graph

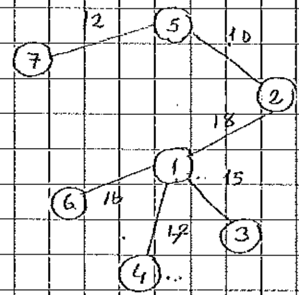
**Question:** Given a connected graph $G=(V,E)$ and a positive set of weights $P_e$ for each edge $e \in E$, find a tree with maximum total edge weight.



$\max_{T} \sum_{e \in T} P_e$ st. T is a tree



84



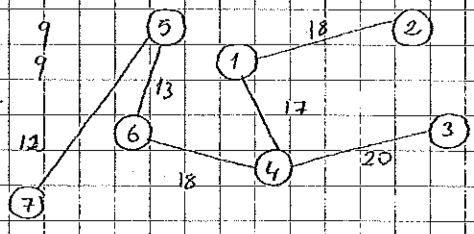
**Greedy Algorithm** (for maxWT)

**Step 0:** Sort all edges of the graph in descending order of edge weights. $T=\{\emptyset\}$

**Step 1:** Go to the top of the sorted list of edges. $L$

If adding $e^*$ to $T$ (current autograph) does not create a cycle then add $e^*$ to $T$ ($T = T \cup \{e^*\}$); if $|T| = n-1$ stop; else go to 1 else remove $e^*$ from $L$, go to 1.

| | | | |
|---|---|---|---|
| (3,4) | 20 ✓ | (1,7) | 9 |
| (4,6) | 18 ✓ | (1,5) | 9 |
| (1,2) | 18 ✓ | | |
| (1,4) | 17 ✓ | | |
| (1,6) | 16 ✓ | | |
| (1,3) | 15 ✓ | | |
| (5,6) | 13 ✓ | | |
| (5,7) | 12 ✓ | | |
| (6,7) | 12 | | |
| (2,3) | 11 | | |
| (3,6) | 11 | | |
| (2,5) | 10 | | |

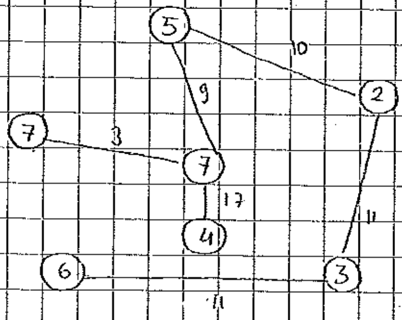


98

GIPTA

Theorem (Prim's Algorithm): The greedy algorithm solves the Max WT Problem.

Greedy Algorithm for a Min WT

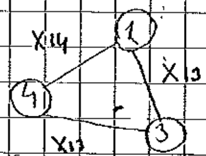



67

LP Formulation for Max WT

Decision variables: $x_e = \begin{cases} 1 & \text{if } e \in T \\ 0 & o/w \end{cases}$ Not enough to prevent cycle.

Max $\sum_{e \in E} P_e x_e$ St $\sum_{e \in E} x_e = n-1$ (min weighted tree)




$C = \{1, 3, 4\}$ $x_{4} + x_{34} + x_{1} \leq 2$

$E(C)$

Max $\sum_{e \in E} P_e t$

St. $(\sum x_e = n-1)$ $\sum_{e \in E(C)} x_e \leq |C| - 1$ $C \subseteq V,$ $|C| \geq 3$

Midterm Question:

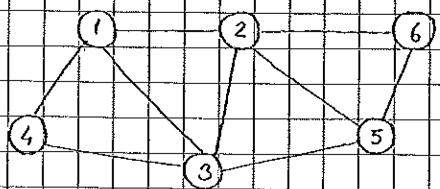
## 1.10 MATCHING AND COVERING PROBLEMS

Given an undirected graph (network) $G = (V, E)$

Problem 1: Find a subset $M$ of edges in $E$ such that the spanning subgraph $G' = (V, M)$, all nodes have degree at most 1

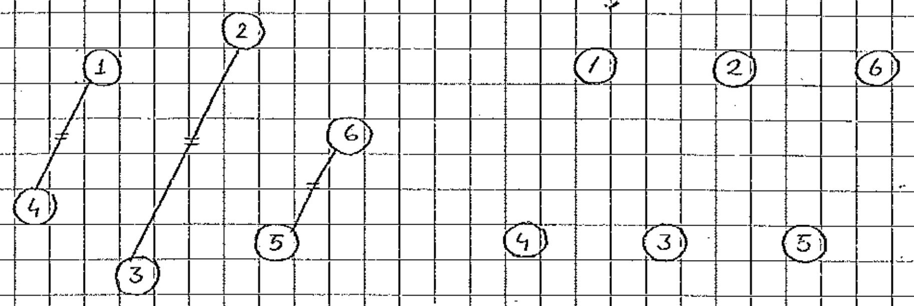
Definition: The degree of a node $i$, $\deg(i)$, in a graph $G = (V, E)$, is the number of edges adjacent to it in $V$.

EX:




$\deg(1) = 3$
$\deg(3) = 4$
⋮

Back to Problem 1



$M = \{(1,4), (2,3), (5,6)\}$

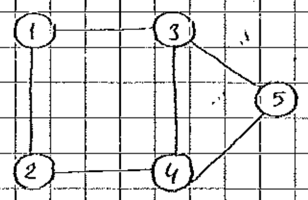
Definition: Such subsets $M$, are called matching.

The empty set is matching. So, in any graph there is a matching.

Problem 2: Find a matching maximum cardinality.

Definition: If in a matching $m$, the resulting subgraph $G = (V, M)$ has all nodes with degree equal 1, $M$ is called a perfect matching.

EX:



MCM: Maximum cardinality matching.

has no perfect matching.

Problem 3: Assume that all edges have positive weights, $C_e > 0$ $\forall e \in E$. Find a matching with the largest total weight.

( Weight of a matching $= \sum_{e \in M} C_e$ )

Observation: Problem 2 is a special case of Problem 3; because if I chose $C_e = 1$ $\forall e \in E$ then Problem 3 reduces to Problem 2.

MWM $\Leftrightarrow$ MCM when all edge weights are identical
$\downarrow$
equivalent

Fact: Problem 2 and 3 can be formulated as integer linear programing problems.

Decision variable: $X_e = \begin{cases} 1 & \text{if edge } e = (i,j) \text{ is in } M. \\ 0 & \text{o/w} \end{cases}$

$$\text{Max} \sum_{e \in E} c_e x_e$$

St

$$\deg(i) \leq 1 \iff \sum_{e \in \delta(i)} x_e \leq 1$$

For $i$, if I know all the edges adjacent to $i$, $\delta(i)$,

So the formulation:

$$\text{Max} \sum_{e \in E} c_e x_e$$

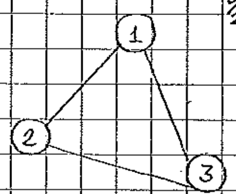
St.

$$\sum_{e \in \delta(i)} x_e \leq 1$$

$$\cancel{x_e \in \{0,1\} \ \forall e \in E} \qquad x_e \geq 0$$

<u>Question</u>: Does formulation with $x_e \geq 0 \ \forall e$ and without $x_e \in \{0,1\} \ \forall e$ give a matching when solved as an LP?

<u>EX</u>: $G = (V,E)$ $c_e = 1 \ \forall e$




$$\tfrac{3}{2} > 1 = \text{Max} \ \overset{1}{x_{12}} + x_{13} + x_{23}$$

$$\text{St.} \quad \begin{aligned} \overset{1}{x_{12}} + x_{13} \quad\quad\quad & \overset{1/2}{\leq} 1 \\ x_{12} \quad\quad + x_{23} & \leq 1 \\ x_{13} + x_{23}^{1/2} & \leq 1 \end{aligned}$$

$$\cancel{\begin{matrix} x_{12} \in \{0,1\} \\ x_{13} \\ x_{23} \end{matrix}} \Rightarrow \begin{matrix} 0 \leq x_{12} \leq 1 \\ 0 \leq x_{13} \leq 1 \\ 0 \leq x_{23} \leq 1 \end{matrix}$$

<u>Answer</u>: NO in general

BUT, YES iff the graph $G$ is <u>bipartite</u>

$G = (V,E)$ such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and all edges $e = (i,j) \in E$ are such that $i \in V_1$ and $j \in V_2$.

So, the MWM (or MCM) can be solved by solving the linear Programming Problem

$$\text{Max} \sum c_e x_e$$

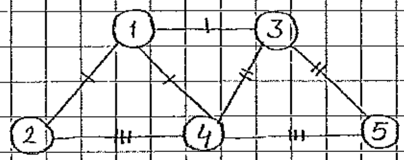
St

$$\sum_{e \in \delta(i)} x_e \leq 1 \ \forall i \in V \qquad x_e \geq 0 \qquad \underline{G \text{ is bipartite}}$$

Covering: Minimum Edge Cover Problem by nodes

In a given graph, chose a subset of nodes such that for each edge $e=(i,j)$, either $i$ or $j$ or both are in the chosen subset of nodes this is called an edge cover by nodes

$G=(V,E)$



$N=\{1,2,3,5\}$ is an edge cover by nodes.

$\mu(G)=1$
$\chi(G)=2$

Problem 4: Among all edge covers by nodes, find one with minimum cardinality called the problem of minimum edge cover by nodes.

Formulation Using Integer Linear Programming

$$y_i=\begin{cases}1 & \text{if } i\in N\\ 0 & \text{o.w}\end{cases} \qquad \forall i\in V$$

$\forall i \in V$

$$\text{Min} \sum_{i\in V} c_i y_i \qquad c_i>0$$

$$y_i+y_j\geq 1 \quad \forall e=(i,j)\in$$

$$\text{Min } y_1+y_2+y_3+y_4+y_5$$

$$\begin{aligned}
\text{St}\quad & y_1+y_2 \geq 1\\
& y_1+y_3 \geq 1\\
& y_1+y_4 \geq 1\\
& y_2+y_4 \geq 1\\
& y_3+y_4 \geq 1\\
& y_3+y_5 \geq 1\\
& y_4+y_5 \geq 1
\end{aligned}$$

$$y_1,y_2,y_3,y_4,y_5\in\{0,1\}$$

$$1\geq y_i\geq 0$$

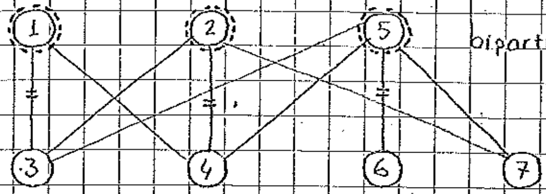
Fact: a) I can pass to the LP without losing the opti ed. c by nodes iff and only if G is bipartitle

b) For any $G$, $\mu(G)\leq \chi(G)$

c) For bipartitle $G$ $\mu(G)=\chi(G)$

GIPTA

EX




bipartitle

$G = (V, E)$

$\mathcal{M}(G) = ?$

$M(G) = 3$

$\chi(G) = 3$

$$\text{Max} \sum_{e \in E} c_e x_e$$

St

$$\sum_{e \in \delta(i)} x_e \le 1 \qquad \forall i \in V$$

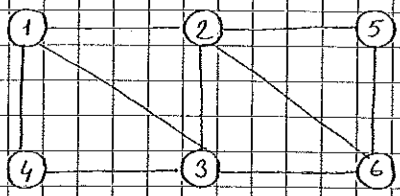
$$\sum_{e \in C} x_e \le \frac{|C|-1}{2}$$

$$\forall C \subseteq V : |C| \text{ is odd}$$

$$C = \{(i,j) \in E : i \in C, j \in C\} \qquad x_e \geq 0$$

If solved as an LP, this problem gives the optimum matching.

EX




Solve the LP version of $\mathcal{M}(G)$ MWM formulation.

(GAMS)

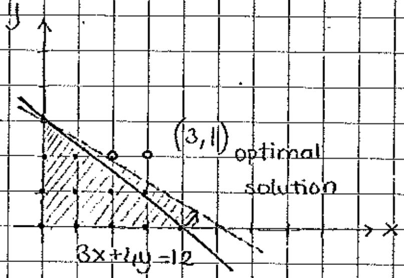
GIPTA

# CHAPTER 2 / INTEGER PROGRAMMING

Integer valued variables give us an extended modeling capability, but at a high cost. Meaning Integer programming is a lot harder than LP.

EX: max $3x + 4y$
St
$$5x + 8y \leq 24$$
$$x, y \geq 0 \quad \cancel{\text{integer}}$$

The first that one does, when faced with an LP, is to solve the linear relaxation (LP Relaxation)




The LP Relaxation gives the optimal solution $x = 24/5$, $y=0$ (obviously does not solve the original problem)

1st attempt: Round up the fractional value $x=5$, $y=0$ (infeasible)

2nd attempt: Truncate the fractional part $x=4$ $y=0$ (feasible) $z = 12$ not optimal

## 2.1 MODELING POWER OF INTEGER VARIABLES

Capital Budgeting

Company XYZ which selects among six different investment proposals. Each project has an initial cash outlay $C_j$, $j = 1, \ldots, 6$ and a present value of $P_j$, $j = 1, \ldots, 6$. The company has a budget of B into these 6 projects.

Question: Which projects should the company chose so as to maximize the resulting net present value?

Decision Variables:

$$x_j = \begin{cases} 1 & \text{if we invest in project } j. \\ 0 & \text{o.w} \end{cases}$$

Objective Function:

$$\text{Max} \sum_{j=1}^{6} P_j x_j \qquad \text{St} \quad \sum_{j=1}^{6} C_j x_j \leq B$$

GIPTA

$c = (5\ 7\ 4\ 3\ 4\ 6)$
$p = (16\ 22\ 12\ 18\ 11\ 19)$

$$\text{Max } 16x_1 + 22x_2 + 12x_3 + 18x_4 + 11x_5 + 19x_6$$
S.t

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

$x_1 + x_3 \leq 1$
$x_1 \leq 1 - x_3$

Company has the further regulations/restrictions

a. At most three projects can be selected
b. If $p_1$ is selected then $P_2$ is also selected.
c. If $p_3$ is selected then $P_1$ cannot be selected
d. Either $P_4$ should be selected or $P_5$ should be selected, but not both one of them exactly.

$$x_4 + x_5 = 1$$

$a_1$ Three projects cannot be selected. $a_2$:

$$x_1 + x_2 + x_3 + x_4 + x_5 \geq 4w$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 2 + 4w$$

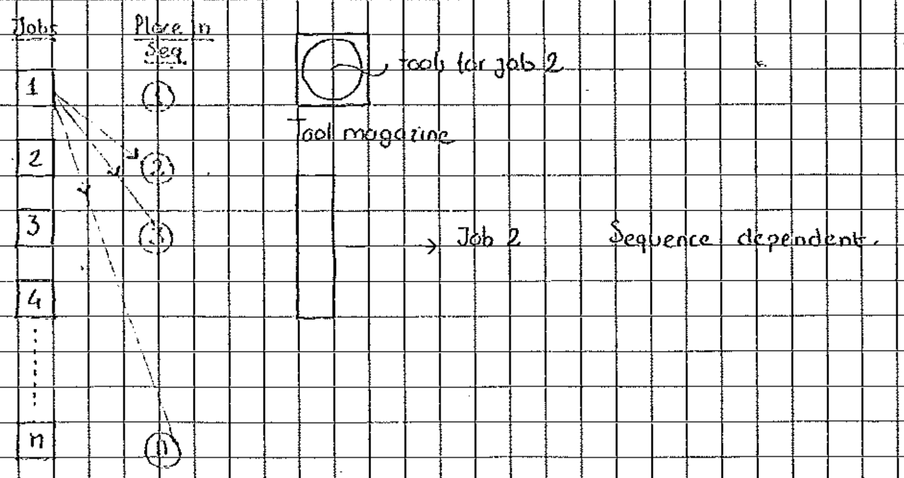
$w \in \{0, 1\}$

e. If $NPV < 42 \Rightarrow x_1 = 1$

$$x_1 \geq (42 - NPV)/42 \quad M (m \geq 42)$$

$$NPV = 16x_1 + 22x_2 + 12x_3 + 18x_4 + 11x_5 + 19x_6$$

! **EX:** Sequence dependent Scheduling of a flexible machine. Machine that can perform m diffrent operations. For each operation, the machine requires a unique tool. The machine has a tool magazine with a total capacity of B (< m). At the begining of each day there are n jobs waiting to be processed by the machine, each job i requires $J_i$ operation on the machine. For a job, begin processing, all tools required by the job have to mounted on the tool magazine. But it takes $s_j$ time units to mount or discount tool j from the tool magazine.



12.10.2009

<u>Question</u>: Find in which sequence the job should be processed by the machine so as to minimize the total time to complete the processing of n jobs.

MODEL

<u>Decision Variables</u>:

$$x_{ir} = \begin{cases} 1 & \text{if job } i \text{ is the } r^{th} \text{ job to be processed } r = 1, \dots, n \\ 0 & \text{O.w} \end{cases}$$

$$y_{jr} = \begin{cases} 1 & \text{if tool } j \text{ is in the tool magazine for the } r^{th} \text{ job in the sequence} \\ 0 & \text{O.w} \end{cases}$$

<u>Constraints</u>:

① Each job should have a position

$$\sum_{r=1}^{n} x_{ir} = 1 \quad \forall i = 1, \dots, n$$

② Each position should be reserved for a single job.

$$\sum_{i=1}^{n} x_{ir} = 1 \quad \forall r = 1, \dots, n$$

③ $x_{ir} \leq y_{jr}$, Job $i$ cannot begin unless its tools are mounted.

$\forall i = 1, \dots, n$
$\forall j \in J_i$
$\forall r$

④ Capacity of tool magazine

$$\sum y_r \leq B \quad \forall r = 1, \dots, n$$

① $x_4 = 1$ $x_5 = 1$, not both

$x_4 \cdot x_5 = 1$

② If $x_2 = 1$ then $x_1 = 1$ $\quad x_2 \leq x_1$

③ If $x_1 = 1$ then $x_3 = 0$

$x_1 + x_3 \leq 1$

You incured a setup cost only if $y_{j,r} \neq y_{j,r-1}$ $\forall j = 1, \dots, n$ for fixed $r$.

<u>Assumption</u> The tool magazine is empty. Means $y_{j0} = 0$ $\forall j = 1, \dots, n$

<u>Objective Function</u>:

$$\text{Min} \sum_{j=1}^{m} \sum_{r=1}^{n} s_j \left| y_{j,r} - y_{j,r-1} \right|$$

S.t

①, ②, ③, ④, ⑤, ⑥

<u>Potential Problem</u> The objective function is not linear. But we can <u>linearize</u> (We can find an equivalent formulation that is linear)

GIPTA

$-Z_{jr} \leq y_{j,r} - y_{j,r-1} \leq Z_{jr} \quad \forall j = 1, \dots, n$ $\quad Z_{jr}$ is a nonnegative var. $Z_{jr} \geq 0$.

2.1 Cont'd

EX: The Game of Fiver:

On a 5x5 board, each square is initially coloured white. Each square serves a button that can be pressed on a square $(i,j)$, then $(i,j)$ and all its neighbours $(i+1,j)$, $(i-1,j)$, $(i,j+1)$, $(i,j-1)$ switch color to red, and vice versa.

j ↑ → i

5x5

| | | | | |
|---|---|---|---|---|
| O | O | O | O | O |
| R | | R | | |
| R | | R | R | |
| R | | R | | |
| | | | | |

Assuming that the board is initially all white, how many button pressures should be made to obtain a board which in all red?

More interesting question: The minimum # of button pressures (which?) to make an initially all-white board all-red?

Decision Variable

$$X_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ is pressed} \\ 0 & \text{o.w} \end{cases}$$

Objective Function

$$\min \sum_{i,j=1,\dots,5} X_{ij}$$

$\forall (i,j) \in B \quad X_{ij} + X_{i-1,j} + X_{i+1,j} + X_{i,j-1} + X_{i,j+1}$ should be an odd number

S.t. $\boxed{W} \xrightarrow{X} \boxed{R}$

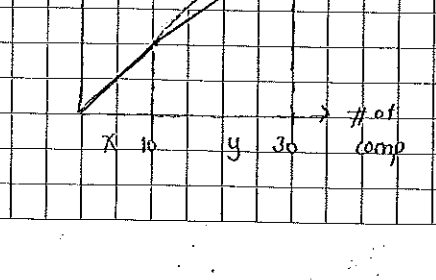
$$X_{ij} + X_{i+1,j} + X_{i-1,j} + X_{i,j-1} + X_{i,j+1} = 2y + 1 \qquad y \in \mathbb{Z}^+ \quad y \in \{0, 1, \dots\}$$

Another use of Integer Variables

EX: We can model non linear cost functions using binary variables. A school is going to buy computers for use in the classrooms. They receive the following offer from their supplier company.

For the first ten computers, they have to pay 2000 TL/computer and for the next 20 computers, the unit price goes down. 1500/computer.

The school wants to maximize the utility derived from computerized teaching which is a linear function of # of computers purchased, but is limited by a budget squeeze.



$200x + 1500y$

Decision Variables

x : # of computers brought in the expensive range.
y : # of computers brought in the discount range

$0 \leq x \leq 10$
$0 \leq y \leq y$

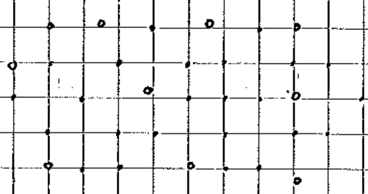
Problem: y cannot begin to take positive values. Before $x = 10$

$w \in \{0,1\}$ Binary variable

$\text{Max } U(x+y) = x+y$

$200x + 1500y \leq B \quad 10 \geq x \geq 10w, \quad 0 \leq y \leq 20w$

Core Problem of Logistics



I: Set of sites on which you can build a warehouse/store

$|I| = m$
$|J| = n$

J = set of customers
$d_j$ = demand of customer j.
$c_{ij}$ = cost of serving customer j from location i
$f_i$ = cost of building a facility at location i.

Fraction of the dem. of J shipped from i

$0 \leq X_{ij}$ = amount shipped from i to j.

$$\sum_{i=1}^{m} X_{ij} = d_j \quad \forall j \in J$$

$$\sum_{j} x_{ij} \leq M y_i \quad \boxed{X_{ij} \leq d_j y_i} \quad \forall i \in I, \; j \in J$$

$$\text{Min } \sum c_{ij} X_{ij} + \sum_{i \in I} f_i y_i$$

$$\sum_{i \in I} \sum_{j \in J} c_{ij} X_{ij} d_j$$

$y_i = \begin{cases} 1 & \text{if Fac is built at loc } i \\ 0 & \text{o.w} \end{cases} \qquad \sum_{i \in I} X_{ij} = 1$ Mixed-Integer Program.

## 2.2 BRANCH AND BOUND METHOD FOR INTEGER PROGRAMMING
(also known as Implicit Enumeration)

**INITIALIZE**: Solve the LP relaxation. If LP is infeasible ⇒ then IP is infeasible STOP else if LP feasible and $x^{LP}$ integer, then stop $x^{LP}$ is an optimal solution in IP else if LP feasible but $x^{LP}$ not integer then $z^{UP} = z^{LP}$, Create two subdivisions of LP, put them in L, and go to SELECT; endif
endif
endif

**CHECK**: If $L = \emptyset$, v $z^{UB}$, then stop; the incumbet is optimal

**SELECT**: If $L = 0$, then stop; the incumbet is optimal else select a subdivision $S$ from L. Solve the resulting linear program (LP)
if infeasible; remove S from L

else if $z^{LP} < z^{LB}$ then remove $S$ from L $\quad z^{UB} = \min(z^{UB}, [z^{LP}])$ (•)

(•) provided that the only subdivision that are alive are the ones just created.

else if $z^{LP} > z^{LB} \wedge x^{LP}$ integer then $x^{I} = x^{LP}$; $z^{LB} = z^{LP}$; $x^{I}$ remov S from L

else if $z^{LP} > z^{LB} \wedge x^{LP}$ not integer; then create subdivisions $z^{LP} < z^{UP}$ of $S$ and then to L; $z^{UB} = \min(z^{UB}, [z^{LP}])$
endif
endif
endif
endif GO TO CHECK

(IP) Max $Cx$
St
$Ax \leq b$
$x \geq 0$
integer

(LP) Relaxation Max $cx$
St
$Ax \leq b$
$x \geq 0$
$\downarrow$ $x^{LP}$

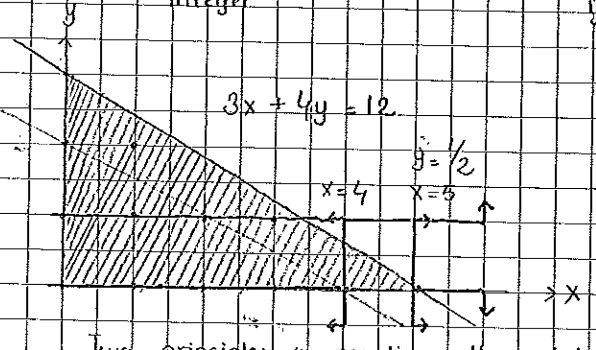
L = list of subproblems that are "alive"

EX: Max $3x + 4y$
$5x + 8y \leq 24$ 12 points
$x \geq 0, y \geq 0$
integer

LP Relaxation

$x = 4.8$ $z^{LP} = \frac{24}{5} \times 3 = \frac{72}{5} = 14.4$ (•)
$y = 0$

(•) an upper bound on the optimal value of IP




$z^* \leq 14.4$ (upper bound)

$z^* \leq \lfloor 14.4 \rfloor = 14$

Two principles in creating the subdivision of the current LP

1. When I solve the two subdivision, I should encounter a fractional solution that I have seen before.

2. I should not leave out any integer feasible solution to IP when I create the subdivisions.

GIPTA

Subdivisions $S1$, $S2$

(S1) Max $3x+4y$

$5x + 8y \leqslant 24$

$x \geqslant 0$

$y \geqslant 0$

$x \geqslant 5$

inf

(S2) Max $3x+4y$

St $5x + 8y \leqslant 24$

$x \geqslant 0$

$y \geqslant 0$

$x \leqslant 4$

$x^{LP} (4, 1/2)$ $\quad Z^{LP} = 12 + 2 = 14$

$y \leqslant 0$ → $S_3$: $y = 0$, $x = 4$

$y \geqslant 1$ → $S_4$: $y = 1$, $x = \frac{16}{5} = 3.2$

$L = \{ S_3, S_4 \}$

$X^I = (4, 0)$

$Z^{LB} = (12)$

$Z^{UB} = \cancel{14}$ $13$

$X^{LP} = (3.2, 1)$

$Z^{LP} = 13.6$

$x \leqslant 3$ → $S_6$: $y = 9/8$, $x = 3$

$x \geqslant 4$ → $S_5$: $x = 4$, $y \geqslant 1$, inf

$L = \{ S_7, S_8 \}$

S6: Max $3x+4y$

$5x + 8y \leqslant 24$

$x \geqslant 0$

$y \geqslant 0, y \geqslant 1$

$x \leqslant 3$

$Z^{LP} = 9 + 4 \cdot \frac{9}{8} = 13.5$

$y = 1$ → $S_7$: $y = 1$, $x = 3$, $Z^{LP} = 13$

$y \geqslant 2$ → $S_8$

new incumbent

$L = \{ S_8 \}$

$X^I = (3, 1)$

$Z^{LB} = 13$

$Z^{UB} = 13$

EX: Max $3x + y$

St $7x + 2y \leqslant 22$

$-2x + 2y \leqslant 1$

$1 \leqslant x \leqslant 4$

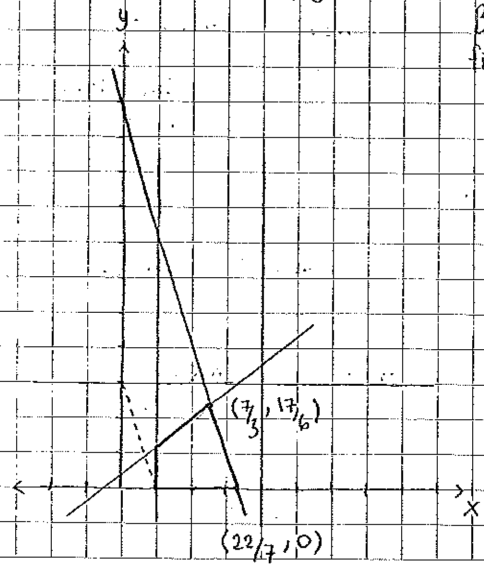
$0 \leqslant y \leqslant 3$

$\pi = 3.14$, $22/7$

$Z^{UB} = 9$

$Z^{LB} = -\infty$

$L = \{S_1, S_2\}$




Breath first

(S)

$x \leqslant 2$ → (S2): $6 + 2.5 = 8.5$, $x = 2$, $y = 5/2$

$x \geqslant 3$ → (S1): $x = 3$, $y = 1/2$, $Z^{LP} = 9.5$

$y \leqslant 0$ → (S3): $y = 0$, $x = \frac{22}{7}$, $Z^{LP} = 9$

$y \geqslant 1$ → (S4): inf

$x \leqslant 3$ → (S6): $x = 3$, $y = 0$ (best integer) solution

$x \geqslant 4$ → (S5): inf

$Z^{UB} = 9$

$Z^{LB} = -\infty$

$L = \{ , S2, S3, S4 \}$

$L = \{ , S2, S5, S6 \}$

$X^I = (3, 0)$

$Z^{LB} = 9$

GIPTA

EX $4x_1 - x_2$

St

$7x_1 - 2x_2 \le 14$

$2x_1 - 2x_2 \le 3$

$0 \le x_2 \le 3$

$0 \le x_1$

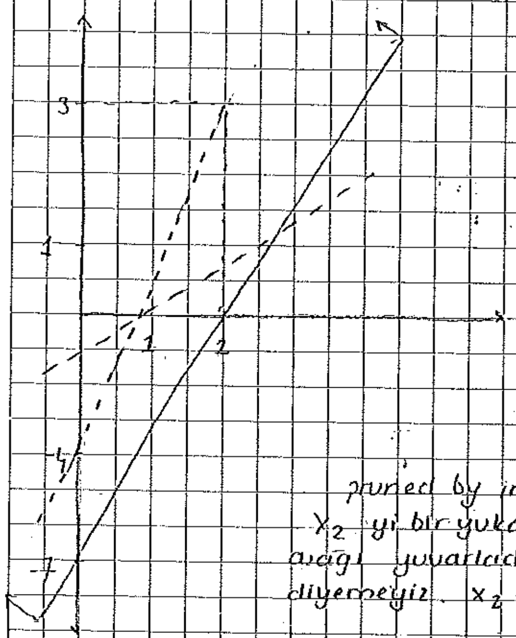
Solve the problem by B-B method

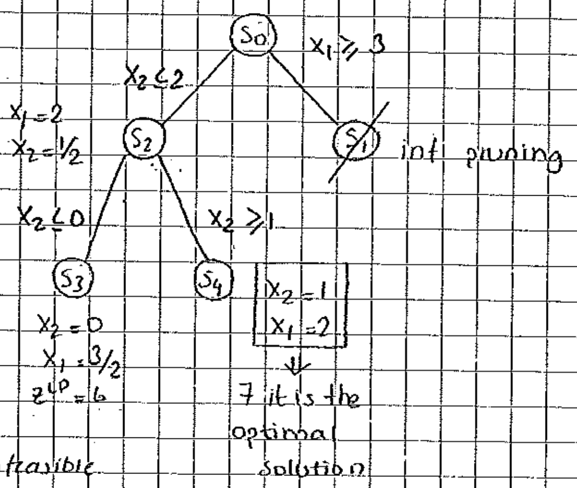
$z = 4x_1 - x_2 = 4$

$z^{LP} = 0$

$z^{UB} = 7$

$z^{LB} = -\infty$




pruned by infeasible

$x_2$ yi bir yukarı bir de aşağı yuvarladı $x_1$ hakkında birşey diyemeyiz. $x_2 = 1/2$ nin olmadığı aralığı yazdık.

EX Max $-x_1 + 4x_2$

St

$-10x_1 + 20x_2 \le 22$

$+5x_1 + 10x_2 \le 49$

$0 \le x_1 \le 5$

$x_2 \ge 0$

integer

Solve the problem by B&B

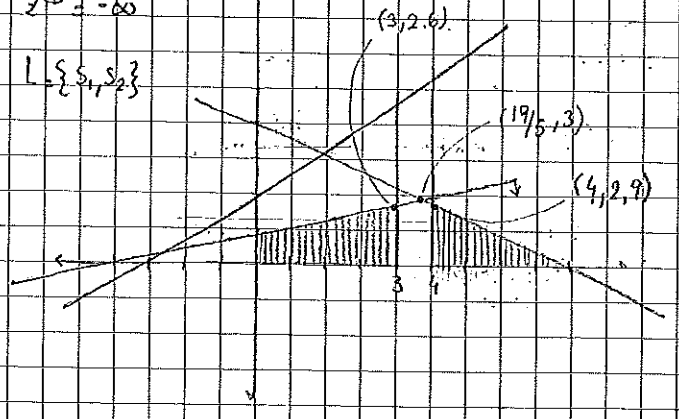
$-x_1 + 4x_2 = 4$

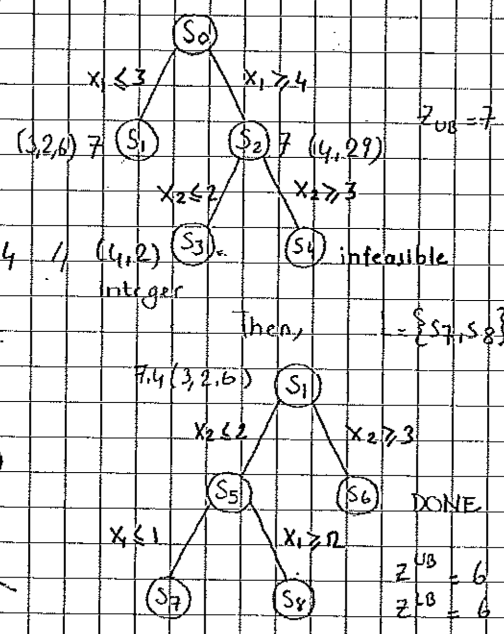
$z^{LP} = 8$

$z^{UB} = 8$

$z^{LB} = -\infty$

$L = \{S_1, S_2\}$




GIPTA

2.2 Cont'd

EX $\max\ 2x_1 + x_2$

s.t.

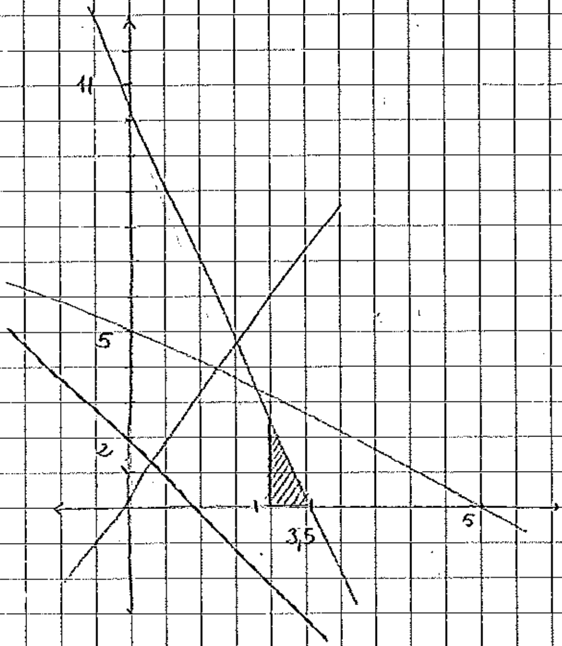
$x_1 + x_2 \leq 5$

$-x_1 + x_2 \leq 0$

$6x_1 + 2x_2 \leq 21$

$x_1, x_2 \geq 0$ integer

Solve by branch and bound.



$x_1 = 11/4 \quad x_2 = 9/4$

$z = 7{,}75$

$x_1 \leq 2$: $S_1$

$x_1 \geq 3$: $S_2$

$S_2$: $x_1 = 3$, $x_2 = 3/2$, $z = 7{,}5$

$x_2 \leq 1$: $S_3$

$x_2 \geq 2$: $S_4$ infeasible

$S_3$: $x_2 = 1$, $x_1 = 3{,}16$, $z = 7{,}32$

$x_1 \leq 3$: $S_5$

$x_1 \geq 4$: $S_6$ infeasible

$S_5$: $x_1 = 3$, $x_2 = 1$, $z = 7$

So, $z^{LB} = z^{UB} = 7$ optimal

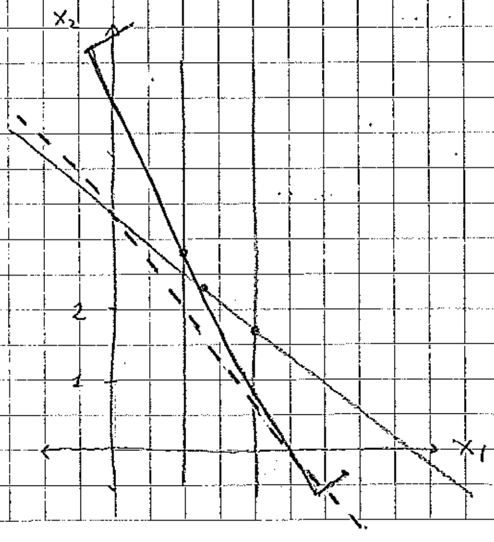
EX $\min\ 15x_1 + 10x_2$

$3x_1 + x_2 \geq 6$

$x_1 + x_2 \geq 3$

$x_1, x_2 \geq 0$ integer

$z^{LB} = 38$ as for LP $z^* = 37{,}5$




$S_0$

$x_1 \leq 1$: $S_1$

$x_1 \geq 2$: $S_2$

$S_1$: $x_1 = 1$, $x_2 = 3$, $z = 45$

$S_2$: $x_1 = 2$, $x_2 = 1$, $z = 40 = z^{UB}$

optimal solution as nothing come out from the node $\gamma$1. capital budgeting 0-1 knapsack problem.

EX max $8x_1 + 11x_2 + 6x_3 + 4x_4$

S.t

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \qquad x_1, x_2, x_3, x_4 \text{ binary}$$

The 0-1 knapsack problem has the following property.

- It's LP relaxation can be solved without using the simplex method.

$$\max \sum_{j=1}^{n} p_j x_j$$

$$\text{St} \quad \sum_{j=1}^{n} c_j x_j \leq b$$

$$x \in \{0,1\}$$

$$0 \leq x_j \leq 1$$

LP relax of

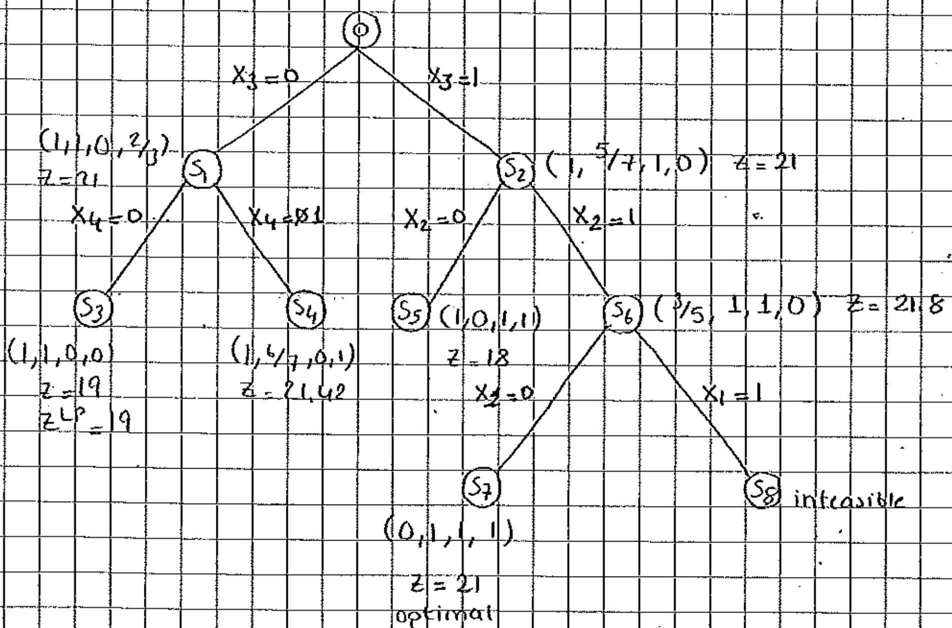
- Greedy Algorithm solves the LP Relaxation of KP to optimality.

1. Compute the ratio. $\frac{p_j}{c_j} \ \forall j = 1, \dots, n$
2. Sort the ratio in descending order.
3. Pick the variable $j$ on top of the list.
   If the current capacity $\bar{b}$ is sufficient then set $x_j = 1$; delete $j$ from the list, $\bar{b} = \bar{b} - c_j$, go back to 3
   else set $x_j = \bar{b}/c_j$ stop
   endif

$8/5 = 1.6 \qquad 11/7 = 1.57 \qquad \underbrace{6/4 = 1.5}_{1/2} \qquad \underbrace{4/3 = 1.33}_{0}$

$X^{LP} = (1, 1, 1/2, 0) \qquad Z^{LP} = 22 = Z^{UB}$

$Z^{LB} = 0$




GIPTA

EX $\max\ 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$

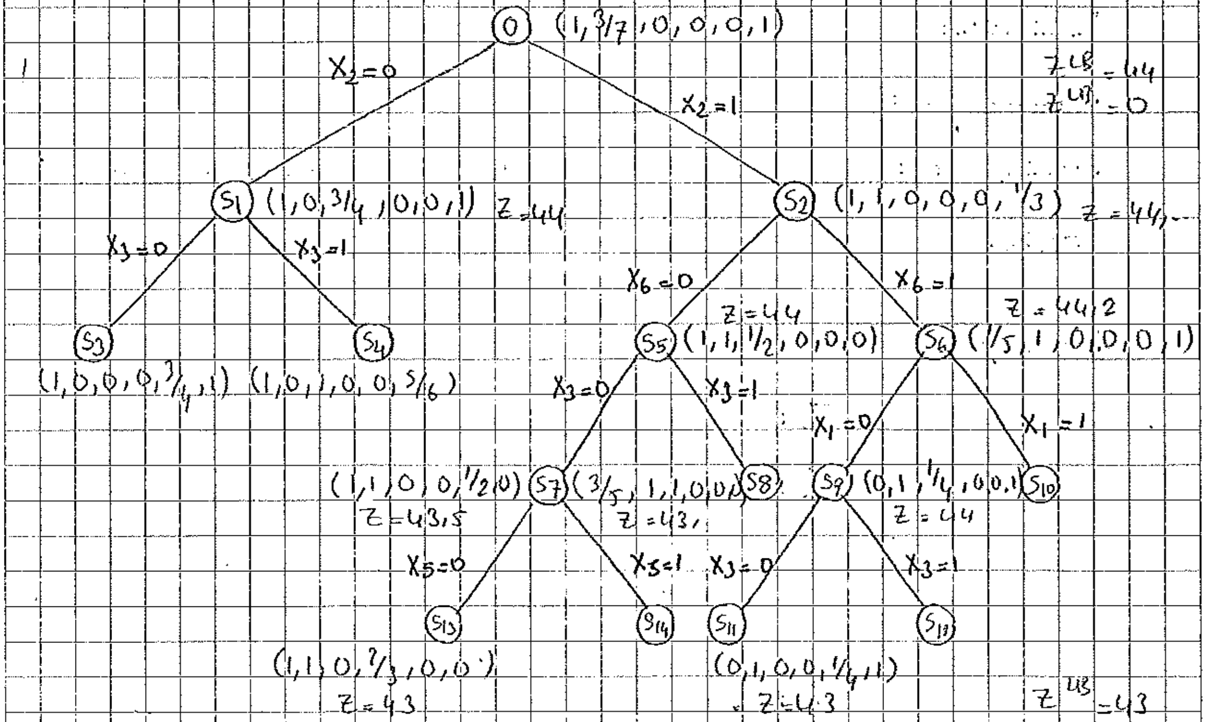
St

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$

$x_i$ binary

| | 1 | 3 | 4 | 6 | 5 | 2 |
|---|---|---|---|---|---|---|
| LP Relax. | $16/5$ | $22/7$ | $12/4$ | $8/3$ | $11/4$ | $19/6$ |
| | = | = | = | = | = | = |
| | 3,2 | 3,14 | 3 | 2,66 | 2,75 | 3,16 |

$( 1, \ 3/7, \ 0, \ 0, \ 0, \ 1 )$



## 2.3 BALAS' ADDITIVE ALGORITHM FOR BINARY INTEGER PROGRAMMING

$$\text{Min} \sum_{J=1}^{n} C_J X_J \qquad \text{(Only min prob)}$$

$$\text{S.t} \quad \sum_{J=1}^{n} a_{iJ} X_J \geq b_i \quad \forall\, i=1,\ldots,m \text{ and } X_J \in \{0,1\} \quad \forall\, J=1,\ldots,n$$

Partition the set of binary solutions to two subsets at each branching step; chose one of the branches, bound and fathom (eliminate that subset from further consideration 3 Fathoming Steps.

Remark: Rearrange (sort) all the variables in ascending order of their objective function coefficient.
2. Let us assume that all variables have positive objective function coefficient.

Initilization: Set $z^{UB} = \infty$, create two subsets, compute a lower bound; go to Branching on the optimal value for each subset.
Branching Step: Select the subset with the smaller lower bound; go to bounding step.
Bounding Step: Apply the three fathoming tests to the current subset. If it fails any of the tests, then take the appropriate action go to stopping.
Stopping: If all subsets have been fathomed stop with the best integer solution obtained so far, else go to branching step.

Bounding:

Fathoming Tests: $T_1$: If $Z_L > Z^{UB}$ fathoming by bound.

$T_2$: If $\exists\, i : \sum_{J=1}^{N} a_{iJ} x_J + \sum_{J=N+1}^{n} \max(0, a_{iJ}) < b_i$

then current subset is infeasible, fathoned by infeasibility.

$T_3$: Immediate feasible completion

Let $X_{N+1} = 1 - X_N$

$Z^{UB} = \min(Z^{UB}$, the value of the infeas completion) $\Leftarrow$ if $(X_1, \ldots, X_N, 1-X_N, 0, 0, \ldots, \overset{n}{0})$ is feasible then fathoned by feasibility.

If all the tests $T_1, T_2, T_3$ are inconclusive, then compute a lower bound for the current potential sequence as follows.

if $X_N = 0$ $\quad Z_L = \sum_{J=1}^{N-1} C_J X_J + C_{N+1}$

$X_N = 1$ $\quad Z_L = \sum_{J=1}^{N} C_J X_J$

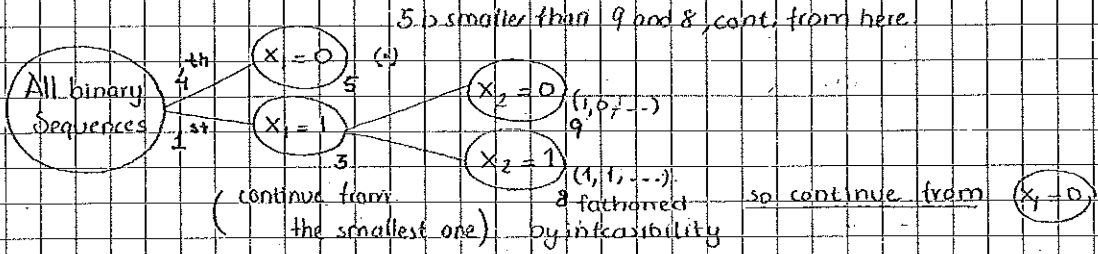
GIPTA

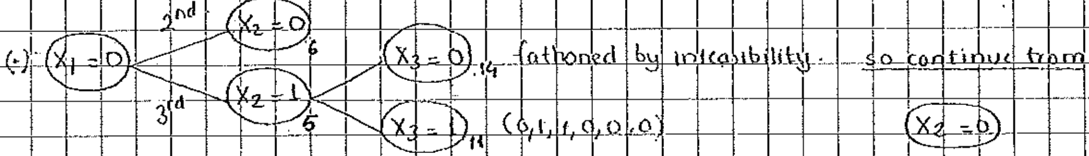
EX Min $3x_1 + 5x_2 + 6x_3 + 9x_4 + 10x_5 + 10x_6$

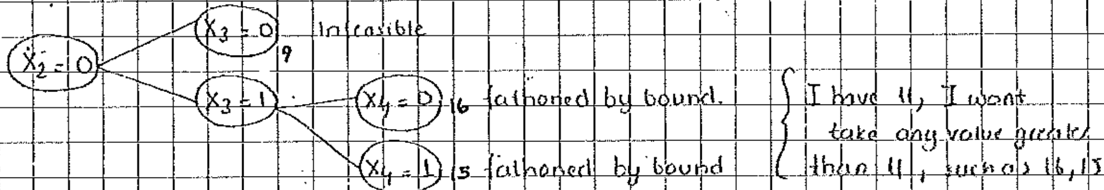
St

$$-2x_1 + 6x_2 - 3x_3 + 4x_4 + x_5 - 2x_6 \geqslant 2$$
$$-5x_1 - 3x_2 + x_3 + 3x_4 - 2x_5 + x_6 \geqslant -2 \qquad x_{1,\dots,6} \in \{0,1\}$$
$$5x_1 - x_2 + 4x_3 - 2x_4 + 2x_5 - x_6 \geqslant 3$$

$Z^{UB} = \infty$



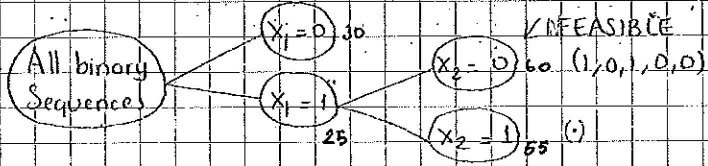



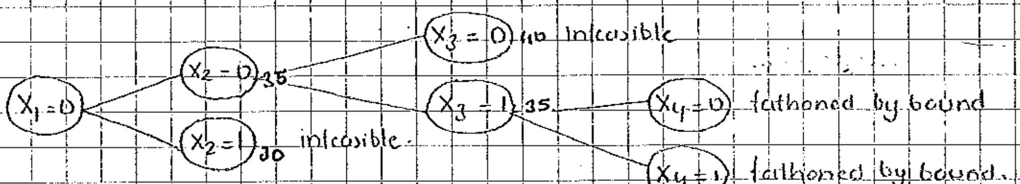



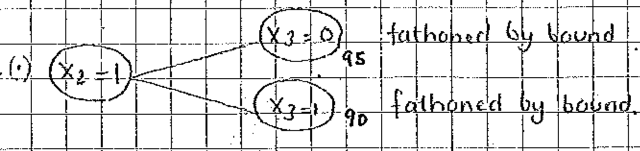

EX Min $25x_1 + 30x_2 + 35x_3 + 40x_4 + 45x_5$

St

$$3x_1 - x_2 + x_3 + x_4 - 2x_5 \geqslant 2$$
$$x_1 + 3x_2 - x_3 - 2x_4 + x_5 \geqslant 0 \qquad x_{1,\dots,5} \in \{0,1\}$$
$$-x_1 - x_2 + 3x_3 + x_4 + x_5 \geqslant 1$$



EX Max $2x_1 - x_2 + 5x_3 - 3x_4 + 4x_5$

St

$$3x_1 - 2x_2 + 7x_3 - 5x_4 + 4x_5 \leq 6$$
$$x_1 - x_2 + 2x_3 - 4x_4 + 2x_5 \leq 0$$

Min $-2x_1 + x_2 - 5x_3 + 3x_4 - 4x_5$

St

$$-3x_1 + 2x_2 - 7x_3 + 5x_4 - 4x_6 \geq -6$$
$$-x_1 + x_2 - 2x_3 + 4x_4 - 2x_5 \geq 0$$

Min $2y_1 + y_2 + 5y_3 + 3y_4 + 4y_5 = -11 + \ldots$

St

$$3y_1 + 2y_2 + 7y_3 + 5y_4 + 4y_5 \geq 8$$
$$y_1 + y_2 + 2y_3 + 4y_4 + 2y_5 \geq 5$$

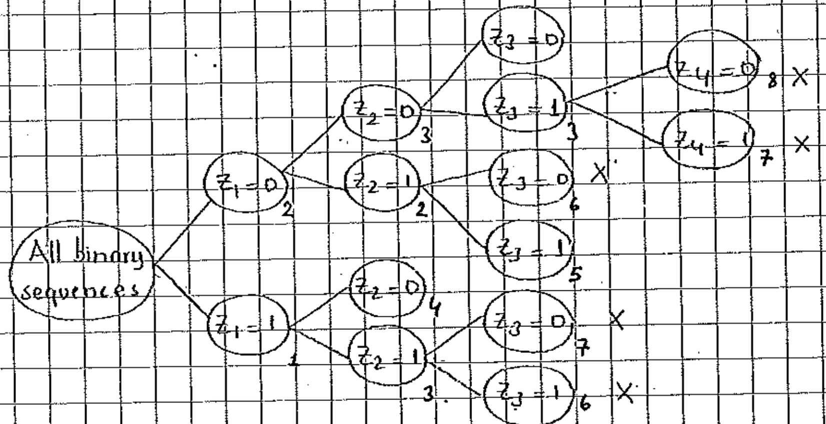
Min $z_1 + 2z_2 + 3z_3 + 4z_4 + 5z_5$

St

$$2z_1 + 3z_2 + 5z_3 + 4z_4 + 7z_5 \geq 8$$
$$z_1 + z_2 + 4z_3 + 2z_4 + 2z_5 \geq 5$$

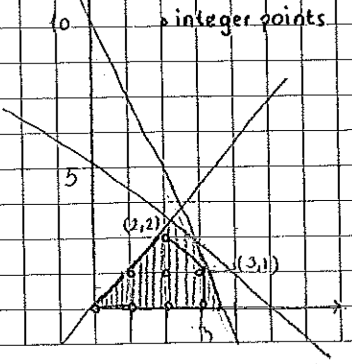
$y_1 = 1 - x_1 \rightarrow x_1 = 1 - y_1$

$y_3 = 1 - x_1$, $y_4 = x_4$, $y_2 = x_2$, $y_5 = 1 - x_5$




## 2.4 IDEAL FORMULATIONS, VALID INEQUALITIES CHVATAL-GOMORY PROCEDURE

In integer programming, we are interested in finding the "best" integer point in a discrete set of integer feasible points X.




Max $2x_1 + x_2$

St

$(P_1)$
$$-x_1 + x_2 \leq 0$$
$$6x_1 + 2x_2 \leq 21$$
$$0 \leq x_1 \quad \text{integer}$$
$$0 \leq x_2$$

$$X = \{(0,0), (1,0), (2,0), (3,0), (1,1), (2,1), (3,1), (2,2)\}$$

discrete set of points

GIPTA

Max $2x_1 + x_2$
St

$$(P_2)\quad \boxed{\begin{array}{l} x_2 \le x_1 \\ x_1 + x_2 \le 4 \\ 0 \le x_1 \le 3 \\ x_2 \geq 0 \end{array}} \xrightarrow{LP} (3,1)$$

$P_2$ has a special name. It is called the convex hull of integer points formulation.
Smallest containing all points of the set $X$.

For the example $P_1$ and $P_2$ are both formulations for the same discrete set $X$ why because all members of $X$ satisfy the inequalities defining $P_1$, $P_2$.

Idea: To find better formulation, as close to the convex hull of integer points as possible.

For the same discrete $X$, let $P_1$ and $P_2$ be two formulations {i.e, $X \le P_1$, $X < P_2$ }

If $P_2 \subset P_1$, then we say that $P_2$ is a better form than $P_1$ tighter.

Max $2x_1 + x_2$
St

$$(P_3)\quad \boxed{\begin{array}{l} x_1 + x_2 \le 5 \\ x_2 \le x_1 \\ 3x_1 + x_2 \le 10 \\ x_2 \geq 0 \\ x_1 \geq 0 \end{array}}$$

$P_2 \subset P_3 \subset P_1$
$\longleftarrow$ Tighter

Definition: A valid inequality $\pi^T x \le \pi_0$, $\pi = \binom{3}{1}$ $\pi_0 = 10$ for a discrete set $X$. is an inequality satisfied by all members of $X$; a strong valid inequality is one that is violated by some fractional points of the current formulation for $X$.

Chvatal - Gomory Procedure.

$a^T x \le b$

$a_1 x_1 + a_2 x_2 + \ldots\ldots + a_n x_n \le b$

$\lfloor a_1 \rfloor x_1 + \lfloor a_2 \rfloor x_2 + \ldots\ldots \lfloor a_n \rfloor x_n \le$

$a_1 x_1 + a_2 x_2 + \ldots\ldots + a_n x_n \le b$

$x_1 \geq 0$
$\vdots$ integer
$x_n \geq 0$

$$\boxed{\lfloor a_1 \rfloor x_1 + \lfloor a_2 \rfloor x_2 + \ldots\ldots + \lfloor a_n \rfloor x_n \le \lfloor b \rfloor}$$

is an inequality valid for all integer values of $x_1, \ldots, x_n$ satisfying !

$EX_1$  $P = \{(x_1, x_2, x_3, x_4) \geq 0,\ 4x_1 + 5x_2 + 9x_3 + 12x_4 \leq 34\}$

The question: Find a V.I by C.G procedure for the set

$$X = P \cap \mathbb{Z}^4$$

Solution: $4x_1 + 5x_2 + 9x_3 + 12x_4 \leq 34$

$$x_1 + \tfrac{5}{4}x_2 + \tfrac{9}{4}x_3 + \tfrac{12}{4}x_4 \leq \tfrac{34}{4}$$

(CG) $\boxed{x_1 + x_2 + 2x_3 + 3x_4 \leq 8}$

$EX_2$  $x_2 + x_3 + 2x_4 \leq 6$ is valid for $X$?

$$\tfrac{4}{5}x_1 + x_2 + \tfrac{9}{5}x_3 + \tfrac{12}{5}x_4 \leq \tfrac{24}{5}$$

$$x_2 + x_3 + 2x_4 \leq 6$$

MIDTERM

## 2.5 CUTTING PLANE ALGORITHMS

Step 0: Solve the LP Relaxation, if solution integer stop

Step 1: Find a Valid inequality that is violated by the current solution; and to the current LP problem and resolve.

Step 2: If solution all integer stop; o.w go back 1

Gomory Cutting Plane Algorithm

Max $Cx$
St
$Ax = b$
$x \geq 0$
($x$ integer)

F ILP  P bounded above and feasible
$\Rightarrow$ there exist an extreme point optimal solution.
An extreme point solution is identified with a basis.

A basis $\mathcal{B}$ is an index set of the set of columns of the matrix $A$, corresponding to $m$ linearly independent columns.

$A = \left.\left[\ \underbrace{| \ | \ | \ \cdots\cdots}_{n}\ \right]\right\} m$   $B$, basis matrix   $\mathcal{B} = \{1, 3, 5\}$ basic variables $x_1\ x_3\ x_5$

$x = (x_B, x_N)$  The equations $Ax = b$ are equivalently expressed as

$$Bx_B + Nx_N = b$$

$\boxed{B^{-1}Bx_B + B^{-1}Nx_N = B^{-1}b}$ , $\boxed{x_B = B^{-1}b - B^{-1}Nx_N}$

A basic solution is obtained by setting $x_N = 0$, $x_B = B^{-1}b$

If $x_B \geq 0$ then it is called a basic feasible solution for (P).

$$cx = c_B^T x_B + c_N x_N$$
$$= c_B^T (B^{-1} b - B^{-1} N x_N) + c_N x_N$$
$$= c_B^T B^{-1} b - c_B^T B^{-1} N x_N + c_N x_N$$
$$z = \underbrace{\underbrace{c_B^T B^{-1} b}_{\text{constant}}}_{\text{value of the current bfs}} + \underbrace{(c_N - c_B^T B^{-1} N)^T}_{\text{reduced costs}} x_N$$

$$Bx_B + Nx_N = b \iff Ax = b$$

$$x_B + B^{-1} N x_N = B^{-1} b \geq 0$$

If the solution is not all integer then there is at least one row $i$ such that $(B^{-1}b)_i$ is fractional

$$(!)\ (x_B)_i + \sum_{j=1}^{n-m} h_{ij} (x_N)_j = \underbrace{(B^{-1}b)_i}_{\text{not an integer}} \geq 0$$

Apply the CG rounding procedure to (!)

$$(x_B)_i + \sum \lfloor h_{ij} \rfloor (x_N)_j \leq \underbrace{(B^{-1}b)_i}_{f_i}$$

x's are integer

$$\boxed{(x_B)_i + \sum \lfloor h_{ij} \rfloor (x_N)_j \leq \lfloor (B^{-1}b)_i \rfloor} \quad (!!)$$

is a valid ineq for all int $x$, $x \geq 0$, $Ax = b$

I know from (!) $(x_B)_i = f_i - \sum_{j=1}^{n-m} h_{ij} (x_N)_j$ substitute this into (!!)

$$f_i - \sum_{j=1}^{n-m} h_{ij} (x_N)_j + \sum_{j=1}^{n-m} \lfloor h_{ij} \rfloor (x_N)_j \leq \lfloor f_i \rfloor$$

Gomory fractional cut:
$$\boxed{\sum_{j=1}^{n-m} (h_{ij} - \lfloor h_{ij} \rfloor) \overbrace{(x_N)_j}^{0} \geq f_i - \lfloor f_i \rfloor} \quad (\cdot)$$
→ greater than 0

$(\cdot)$ is violated by the current solution

EX  Max $2x_1 + x_2$

St

$$x_1 + x_2 \le 5$$
$$-x_1 + x_2 \le 0$$
$$6x_1 + 2x_2 \le 21$$
$$x_1, x_2 \ge 0$$
integer

$x_1 = 11/4 \quad x_2 = 9/4$

$S_1 = 0$

$-11/4 + 9/4 + 1/2 \; S_2$

$S_3 = 0$

$B = \{1, 2, 4\}$

$$B = \left|\begin{array}{ccc|ccc} 1 & 1 & 0 & -1/2 & 0 & 1/4 \\ -1 & 1 & 1 & 3/2 & 0 & -1/4 \\ 6 & 2 & 0 & -2 & 1 & 1/2 \end{array}\right|$$

$\underbrace{\qquad\qquad}_{B^{-1}}$

$$B^{-1}b = \begin{pmatrix} 11/4 \\ 9/4 \\ 1/2 \end{pmatrix}^{(c)} \Rightarrow \quad B^{-1}N = \begin{pmatrix} -1/2 & 0 & 1/4 \\ 3/2 & 0 & -1/4 \\ -2 & 1 & 1/2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{array}{cc} S_1 & S_3 \\ \begin{pmatrix} -1/2 & 1/4 \\ 3/2 & -1/4 \\ 2 & 1/2 \end{pmatrix} \end{array}$$

Row 1: $x_1 - 1/2\, s_1 + 1/4\, s_3 = 11/4^{(c)}$

$$\boxed{\frac{1}{2} s_1 + \frac{1}{4} s_3 \ge 3/4} \quad (GC1)$$

$1/2 - (-1)$

$s_1 = 5 - x_1 - x_2$

$s_3 = 21 - 6x_1 - 2x_2$

If we put $s_1, s_2$ in 3 inequalities we'll obtain original variables.

Row 2: $x_2 + 3/2\, s_1 - 1/4\, s_3 = 9/4$

$$\boxed{1/2\, s_1 + 3/4\, s_3 \ge 1/4} \quad (GC2)$$

$\frac{1}{4} - (a) = \frac{1}{4}$

$a = 0$

Row 3: $s_2 - 2s_1 + 1/2\, s_3 = 1/2$

$$\boxed{1/2\, s_3 \ge 1/2} \quad (GC3)$$

$\frac{1}{2} - (a) = \frac{1}{2}$

$a = 1$

$-1/2 - (-1)$

$GC1 \Rightarrow 1/2(\quad -x_2) + 1/4(\quad -2x_1) \ge -7$

$2x_1 + 4/2\, x_2 \le 7$

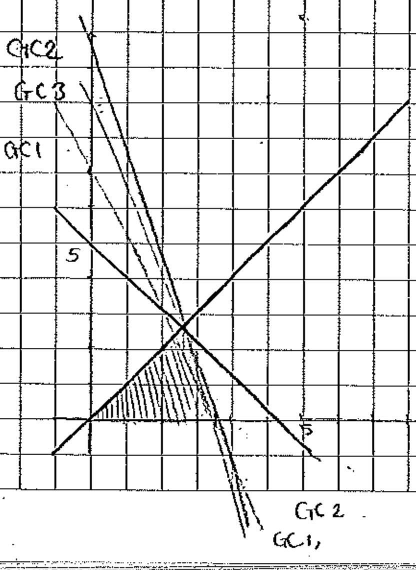
$GC2 \Rightarrow 1/2(5 - x_1 - x_2) + 3/4(21 - 6x_1 - 2x_2) \ge 18$

$5x_1 + 2x_2 \le 18$

$GC3 \Rightarrow 1/2(\quad -6x_1 - 2x_2) \ge -10$

$3x_1 + x_2 \le 10$

GC1 is stronger than GC2.
They do not cut integer points.




GIPTA

## Cutting Plane Algorithm (Cont'd)

For 0-1 Knapsnack Problem

0. Solve the LP Relaxation, Check if solution is integer

Remember 0-1 Knapsack

$$\text{Max } px \quad \text{St } a^T x \le b \quad x \in \{0,1\} \quad \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

if so stop

1. Identify a cutting plane; add it to the current problem.
2. Check if integer if so stop else go back to 1.

For the knapsack problem, the most interesting (usefull) cutting planes are the so called "minimal cover" inequalities.

EX Max $16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$

St

(1) $5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \le 14$

$x_1 + x_2 + x_6 \le 14$

$x_1, \dots, x_6 \in \{0,1\}$

$14-5=9$
$9-6=3$

$X = \{(x_1, \dots, x_t) \in [0,1]$ and satisfying (1) $\} \cap \mathbb{Z}^6$

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| | 1 | 3 | 4 | 6 | 5 | 2 |
| | $16/5$ | $22/7$ | $12/4$ | $8/3$ | $11/4$ | $19/6$ |
| | 3.2 | 3.14 | 3 | 2.67 | 2.75 | 3.16 |

LP Relax:
$x_1 = 1,\ x_6 = 1,\ x_2 = 3/7$
$z^{LP} = 44.42$
$z^{UB} = 44$ aşağı yuvarla

$$z = \text{Max } x_1 + x_2 + x_6$$
St
$$5x_1 + 7x_2 + 6x_6 \le 14 \qquad x_1, x_2, x_6 \in \{0,1\}$$

$x_1 + x_2 + x_6 \le 2$ is valid for X add this to problem

For added model, new LP Relaxation

$x_1 = x_6 = 1,\ x_3 = 3/4,\ z^{LP} = 41$

$$z = \text{Max } x_1 + x_3 + x_6$$
St
$$5x_1 + 4x_3 + 6x_6 \le 14 \qquad x_1, x_3, x_6 \in \{0,1\}$$

$x_1 + x_3 + x_6 \le 2$ add this to the problem as a const.

New LP Relaxation; $x_1 = 1 = x_6,\ x_5 = 3/4$

$x_1 + x_5 + x_6 \le 2$ (add to the problem) ($x_1 + x_2 + x_3 + x_6 \le 2$ also VI) not a min. cover stronger any of these its called extended cover.

New LP Relaxation; $x_1 = 1,\ x_6 = 1,\ x_4 = 1 \quad z^{LP} = 43$

integer solution STOP!

**Definition**: For a 0-1 Knapsack problem, Let $C = \{j : \sum a_j > b\}$
$C'$ is called a cover.

All inequalities of the form.

$$\sum_{j \in C} x_j \leq |C| - 1 \text{ is valid for } X.$$

**Definition**: A minimal cover is a cover $C$ s.t

$$\sum_{j \in C} a_j > b \quad \text{and} \quad \sum_{j \in C \setminus \{j'\}} a_j \leq b \quad \forall j \in C$$

With minimal covers, the knapsack is solved faster using the cutting plane algorithm.

EX. $P = \{(x_1, \dots, x_7) \in [0,1] : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$ $X = P \cap \mathbb{Z}^7$

Question: Find two minimal cover inequalities valid for X

$x_1 + x_2 + x_3 \leq 2$
$x_2 + x_3 + x_4 + x_5 \leq 3$

NOT IN EXAM
(*) How to obtain an extended cover inequality from a min cover ineq?

Take $C$ as a minimal cover and $C' \leftarrow C \cup \{j\}$ where $j'$ st $a_{j'} > a_j$ where $\forall j \in C$.
The resulting inequality is an extended cover inequality. Extended cover inequalities dominate minimal cover inequalities.

extended cover inequality: $\boxed{\sum_{j \in C \cup \{j'\}} x_j \leq |C| - 1}$

## 2.6 IMPORTANT FORMULATIONS REQUIRING CUTTING PLANES (EXPONENTIALLY MANY CONSTRAINTS)

① Optimum tree $G = (V, E)$ $c_e > 0$

Min/Max $\sum c_e x_e$
St $\sum_{e \in E(C)} x_e \leq |C| - 1$

$E(C)$ = set of all edges $e = (i,j)$ s.t $i \in C$ & $j \in C$.
$C \subseteq V$ $|C| \geq 3$ $|V| = n$

② Max Cardinality Matching on $G(V, E)$

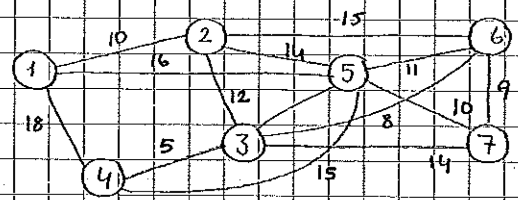
Max $\sum x_e$
St $\sum_{e \text{ adj to } i} x_e \leq 1$ $\forall i \in V$ $\cancel{x_e \in \{0,1\}}$ $x_e \geq 0$ $\forall e$

add cycle constraint $\sum x_e \leq \frac{|C| - 1}{2}$ $\forall C \subseteq V$, add cycle $|C|$ odd and $|C| \geq 3$

## Traveling Salesperson Problem

**Definition:** Given a graph $G=(V,E)$, find whether it contains a path that starts at some node, visits every other node exactly once, and returns to the starting node (Hamilton Tour (cycle)).

EX



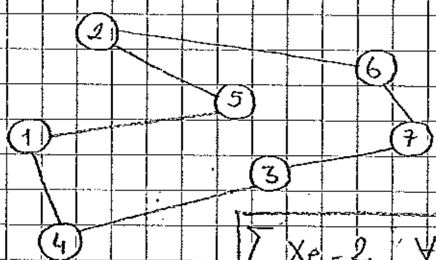
$1-5-2-6-7-3-4-1$ (Hamilton)

$16+14+15+9+14+5+18=91$

**Definition:** Given $G=(V,E)$ with positive $C_e > 0 \ \forall e \in E$ find a shortest Hamilton Tour.

The symmetric Travelling Salesperson Problem is just the shortest Hamilton tour problem.

$$\forall (i,j) \quad C_{ij} = C_{ji}$$



Each node has a degree 2 in a tour.

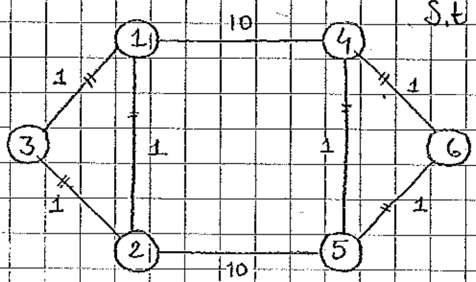
$$\text{Min} \sum_{e \in E} C_e X_e$$

$$X_e = \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{o/w} \end{cases}$$

$\boxed{\sum_{e \text{ adj to } i} X_e = 2 \quad \forall i \in V}$ 2 matching const.

$\boxed{\sum_{i \in S} \sum_{j \in V \setminus S} X_{e=ij} \geq 2 \quad S,\ S \subseteq V,\ |S| \geq 3}$ Cut constraints.

EX $G=(V,E)$



$\text{Min } X_{12}+X_{13}+X_{23}+10X_{14}+10X_{25}+X_{46}+X_{45}+X_{56}$

S.t

$X_{13}+X_{12}+X_{14}=2$

$X_{23}+X_{12}+X_{25}=2$

$X_{23}+X_{13}=2$

$X_{14}+X_{45}+X_{46}=2$

$X_{45}+X_{56}+X_{25}=2$

$X_{46}+X_{56}=2$

$x \in \{0,1\}$

* (2 Matching constraint doesn't satisfy IP solution alone.)

$X_{12}=X_{13}=X_{23}=X_{45}=X_{46}=X_{56}=1$

$X_{14}=X_{25}=0$

That's why we add to formulation a new constraint.

$X_{14}+X_{25} \geq 2$

Let $S=\{1,2,3\}$

Symetric TSP cont'd

**Recall:** Any TSP tour is a 2-matching

A 2 matching $M^2$ in an undirected graph $G=(V,E)$ is a subset of E such that all nodes in V have exactly degree 2 in $M^2$.

$\delta(i)$ = Set of edges adjacent to n.
$E(S)$ = ______ $e=(i,j)$ where $i\in S$, $j\in S$
$S\subseteq V$

$$Z_{M^2} = \text{Min} \sum_{e\in E} c_e x_e \qquad x_{ij} = \begin{cases} 1 & \text{if } e=(i,j)\in T \\ 0 & \text{o/w} \end{cases}$$

$$(3) \quad \sum_{e\in\delta(i)} x_e = 2 \quad \forall i\in V \qquad \forall i\in V$$
$e\in\delta(i)$ i.e. $e$ is adjacent to $i$.

(Two matching) + (5) or (4) = TSP

(1) + (2) + (3) = TSP ($1$-Tree)
ignore $1^{st}$ node then connect it to the tree.

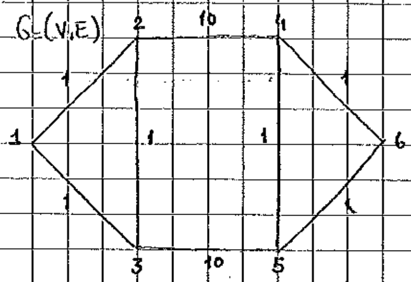
- Cut constraints or equivalent sub tour elimination (4 or 5)

$$(4) \quad \sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 2 \quad \forall S\subseteq V \qquad 1\leq |S| \leq \lfloor V/2 \rfloor$$

$$(5) \quad \sum_{e\in E(U)} x_{ij} \leq |U|-1 \quad \forall U\subseteq V \quad |U|\geq 3$$

As much as any 2-matching is not neccesarily a tour, any 1-tree is not neccesarily a tour.

EX $G(V,E)$




(1 tree)

$$\text{Min} \sum_{e\in E} c_e x_e = Z_{1T} \quad \text{Formulate the min total lenght 1-three}$$

$$x_e = \begin{cases} 1 & \text{if } e\in 1\text{-}T \\ 0 & \text{o/w} \end{cases}$$

$$(1) \quad \sum_{e\in E(U)} x_e \leq |U|-1 \qquad U\subseteq V\setminus\{1\},\ |U|\geq 3, \text{ U cycle.}$$

$$(2) \quad \sum_{e\in E} x_e = |V|$$

**THEOREM:** A graph T is a tour iff T is a two matching and a 1 tree on $G=(V,E)$

Another formulation for STSP:

$$\text{Min} \sum c_e x_e$$
St
$$(1)+(2)+(3) \qquad x_e \in \{0,1\} \quad \forall e\in E \qquad (0)$$

Three formulation all equivalent to one another:

(F1) $\text{Min} \sum_{e\in E} c_e x_e$ s.t (0), (3), (4)

(F2) ______ st (0), (3), (5)

(F3) ______ st (0), (1), (2), (3)

} $Z_{TSP}$

GIPTA

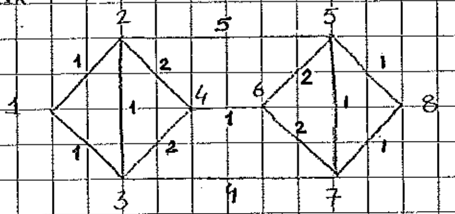
Relaxations of the STSP : A problem with enlarged feasible set whose optimal value is a lower bound on the lenght of the optimal TSP tour.

$Z_{FM^2} \leq Z_{M^2} \leq Z_{TSP}$

Fractional 2 match relax

$Z_{1-T} \leq Z_{TSP}$ 1 tree relaxation of STSP

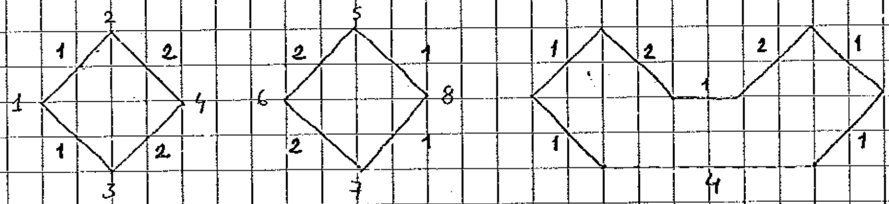
EX: (FINAL)



$Z_{TSP} \geq 12$

$Z_{TSP} \geq 10$

$Z_{M^2} = 12$ $Z_{1-T} = 10$ Compute $Z_{FM}$, $Z_{IM}$



$Z_{1-T} = 9$

$Z_{IM^2} < Z_{1-T} < Z_{FM^2} < Z_{M^2} < Z_{TSP}$

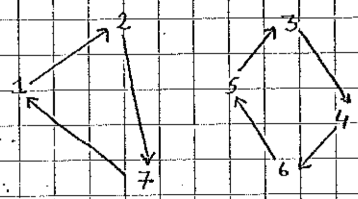
## Asymmetric TSP

$\text{Min} \sum_{(i,j) \in A} d_{ij} x_{ij}$, $d_{ij} \neq d_{ji}$ order matters (Sym case $d_{ij} = d_{ji}$ $\forall (i,j)$)

$x_{ij} = \begin{cases} 1 & \text{if immediate procedures } j \text{ in the tour.} \\ 0 & \text{o/w.} \end{cases}$

$\sum_{i \in \delta^-(j)} x_{ij} = 1$ $\delta^-(j)$ = all nodes $k$ such that $\exists (k,j) \in A$

$\delta^+(j) = \exists (j,k) \in A$



Assignment Constraints

$\sum_{i \in \delta^-(j)} x_{ij} = 1$ (1)

$\sum_{k \in \delta^+(j)} x_{jk} = 1$ (2)

+ Cut constraints $\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1$ (3) $\emptyset \neq S \subset V$

$F1_{ATSP} = \text{Min} \sum d_{ij} x_{ij}$ (0), (1), (2), (3)

Without (3), You have the assign relax of ATSP

## CHAPTER 3 / HEURISTICS

Q: What is a heuristic?
A: Heuristic is an algorithm that does not gurantee an optimal solution, but its simple and "fast."

Heuristics and Exact Algorithms

LP Simplex method
IP Branch & Bound
CPA of Gomory

} They produce an optimal solution

Q: How to measure the speed of algorithms?
A: Define the size of a problem P

size(P) ~ measure of the size of the problem

e.g $G=(V,E)$ $|V|=n$, $|E|=m$, $|A|=m \Rightarrow \boxed{size(P)=n+m}$

(min spanning tree, or max cardinality matching, or shortest path, or max flow)

Min $cx$ (LP)
$Ax=b$
Size (LP) = $\underbrace{m}_{\#\text{ variables}} + \underbrace{n}_{\#\text{ constraints}}$

We say that an algorithm runs in polynomial time if the # of operations used by the algorithm runs in polynomial time if the # of operations, $\mp$, $*$, comparisons used by the algorithm on a problem P of size, size(P), until optimality is a funct that is bounded above by a polynomial in size P.

Algorithm A on P, size(P)

# of operations of Alg A on $P \leq K$. (polynomial in size (P))

e.g size$(P)^3$ + size$(P)^2$ + ...

Greedy Algorithm sort; $m^2+m$ $O(m^2)$ algorithm

A has complexity bound $O(size(P)^3)$

Heuristics will run in poly time.

$P \longrightarrow \boxed{\text{HEURISTIC}} \longrightarrow$ Solution is not necesarily optimal.

Our heuristic in Min problem this range.
$2z^*$
$z_{four}$
$z^*$

For a given heuristic algorithm H, $Z_H \geqslant Z_{OPT}$ (for a min)
Can you say $Z_H \leq K^2 Z_{opt}$, $K \geqslant 1$ performance gurantee

GIPTA

## 3.1 0-1 KNAPSACK PROBLEM NPCOMPLETE

$$\begin{bmatrix} \text{Max} & \sum_{j=1}^{n} p_j x_j & \\ \text{St} & \sum_{j=1}^{n} w_j x_j \leq C & x_j \in \{0,1\} \ \forall j \in \{1,\dots,n\} \end{bmatrix}$$

### A Greedy Heuristic for Knapsack

Sort this items in descending order of $p_j / w_j$ (break ties lexicographically)

Fill the knapsack until the critical item is reached; $j^*$ such that

$w_j > \bar{C}$ (residual capacity) $\Rightarrow \boxed{C - \sum_{j=1}^{s} w_j}$

$x_{j^*} = 0$, all other remaining variables to zero.

**EX:** $p = (15 \ 100 \ 90 \ 60 \ 40 \ 15 \ 10 \ 1)$
$w = (2 \ 20 \ 20 \ 30 \ 40 \ 40 \ 60 \ 10)$ $\quad c = 102$

$7.5 \quad 5 \quad 4.5 \quad 2 \quad 1 \quad 3/8 \quad 1/6 \quad 1/10$

$(1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0), (1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1)$

$Z_G^1 = 15 + 100 + 90 + 60 = 265 \qquad Z_G^2 = 266 \qquad Z_{opt}$ ? (In GAMS)

$\dfrac{Z_G}{Z_{opt}} \to 0 \qquad \dfrac{Z_{opt}}{Z_G} \to +\infty$

**EX:** Max $x_1 + k x_2$
$x_1 + k x_2 \leq k \quad k > 1 \qquad x_1 \in \{0,1\},\ x_2 \in \{0,1\}$

$X_G = (1,0) \quad Z_G = 1$

$X_{opt} = (0,1) \quad Z_{opt} = K \qquad \dfrac{Z_G}{Z_{opt}} = \dfrac{1}{k} \to 0 \quad k \to +\infty$

**EX:** $p = (15 \ 100 \ 90 \ 60 \ 40 \ 15 \ 10 \ 1)$
$w = (2 \ 20 \ 20 \ 30 \ 40 \ 30 \ 60 \ 10)$ $\quad c = 102$

$7.5 \quad 5 \quad 4.5 \quad 2 \quad 1 \quad 1/2 \quad 1/6 \quad 1/10$

$(1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0) \quad Z_G^1 = 265$
$(1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0) \quad Z_G^2 = 280$

## 3.2 THE SYMMETRIC TSP

$G=(V,E)$ $c_E > 0$ $\forall e \in E$ Find a tour of minimal lenght
$c_{ij} = c_{ji}$

1. Greedy
2. Greedy Feasible
3. Nearest insertion
4. K Interchange

the first 3 algorithms are constructive, (start from an empty set and finish with a tour.)

$4^{th}$ one improvement heuristics. (starts with a tour and tries to improve it)

**1. Greedy Heuristic:** Pick a node i to start from; find its nearest neighbour j and add (ij) to T; Repeat for j until T is a tour.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 96 | 105 | 53 | 41 | 86 | 46 | 29 | 56 | 70 |
| 2 | | | 78 | 49 | 94 | 21 | 64 | 63 | 41 | 37 |
| 3 | | | | 60 | 84 | 61 | 54 | 86 | 76 | 51 |
| 4 | | | | | 45 | 35 | 20 | 26 | 17 | 18 |
| 5 | | | | | | 80 | 36 | 55 | 59 | 64 |
| 6 | | | | | | | 46 | 50 | 28 | 8 |
| 7 | | | | | | | | 45 | 37 | 30 |
| 8 | | | | | | | | | 21 | 45 |
| 9 | | | | | | | | | | 25 |

403: $1 \xrightarrow{29} 8 \xrightarrow{21} 9 \xrightarrow{17} 4 \xrightarrow{18} 10 \xrightarrow{8} 6 \xrightarrow{21} 2 \xrightarrow{64} 7 \xrightarrow{36} 5 \xrightarrow{84} 3 \xrightarrow{105} 1$

384: $5 \xrightarrow{36} 7 \xrightarrow{20} 4 \xrightarrow{17} 9 \xrightarrow{21} 8 \xrightarrow{29} 1 \xrightarrow{70} 10 \xrightarrow{8} 6 \xrightarrow{21} 2 \xrightarrow{78} 3 \xrightarrow{84} 5$

**2. Greedy Feasible:** Start $E^f = \emptyset$

form G by adding to $E^f$ the edge with the smallest weight as long as

$|E^f| < n$

and

all nodes in G have degree less than or equal to 2.

and no subtours are formed

EX

$6 \xrightarrow{8} 10 \xrightarrow{18} 4 \xrightarrow{17} 9 \xrightarrow{21} 8$

$6 \xrightarrow{21} 2$, $8 \xrightarrow{29} 1$

$2 \xrightarrow{78} 3 \xrightarrow{54} 7 \xrightarrow{36} 5 \xrightarrow{41} 1$

323

## Heuristics for STSP (cont'd)

We have seen
1) Greedy
2) Greedy feasible
3) Nearest insertion
4) 2-opt heuristic

1)–3) } Constructive Heuristics
4) } improvement heuristics

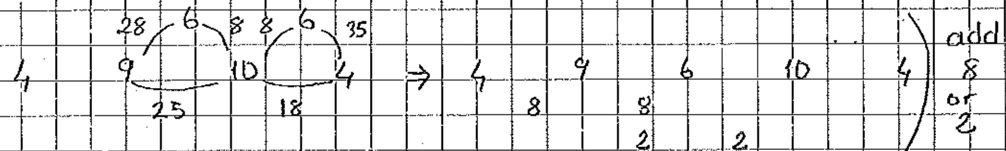
### 3. Nearest Insertion :

STEP 0: Start with a subtour of three nodes (Farthest three nodes to one another)

STEP 1: Find the node that is closest to one of the nodes in the current subtour; insert this node into the subtour to create a new subtour.

STEP 2: If all nodes have been visited ; stop o/w go back 1.

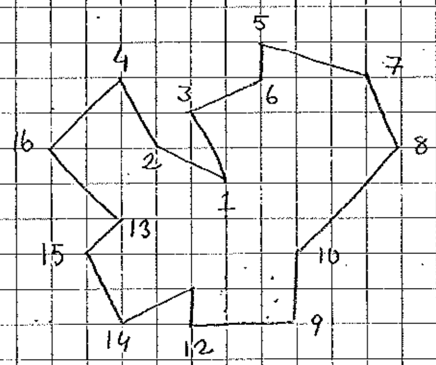
EX look at the previous example's data

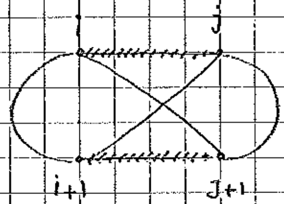



continue like this until you obtain a complete tour.

### 4. 2-Opt Heuristic

Find two edges in the current tour that can be replaced by the two other edges that are not currently in the tour. If this replacement leeds to a shorter tour; implement the replacement.




////// , deleted $(i,j)$ , $(i+1, j+1)$

— , added $(i, j+1)$ , $(j, i+1)$ .

## 3.3 HEURISTICS FOR COVERING PROBLEMS

We have already studied the MECN (min weighted edge cover by nodes)

Given a graph $G=(V,E)$ with positive node weights $\omega_i$, $\forall i \in V$ find a subset $C \subseteq V$ st $\forall (i,j) \in E$ either $i$ or $j$ are in $C$. This is called covering of edges by nodes. The MECN consists finding the cover of smallest total weight.

$$\begin{aligned} \text{Min} \; & \sum_{i \in V} \omega_i y_i \\ \text{St} \; & \\ & y_i + y_j \geq 1 \quad \forall (i,j) \in E \\ & y_i \in \{0,1\} \quad \forall i \in V \end{aligned}$$

Belongs to the same class as TSP (NP complete type problem)

btw our heuristic soln. $\left\{ \begin{array}{l} K^2 \, Z_{opt} \\ \\ Z_{opt} \end{array} \right.$

Greedy Heuristics for MECN

STEP 0: For each node $i$ compute the ratio $C \neq 0$

$$f_i = \frac{\omega_i}{\#\text{ of edges incident to } i}$$

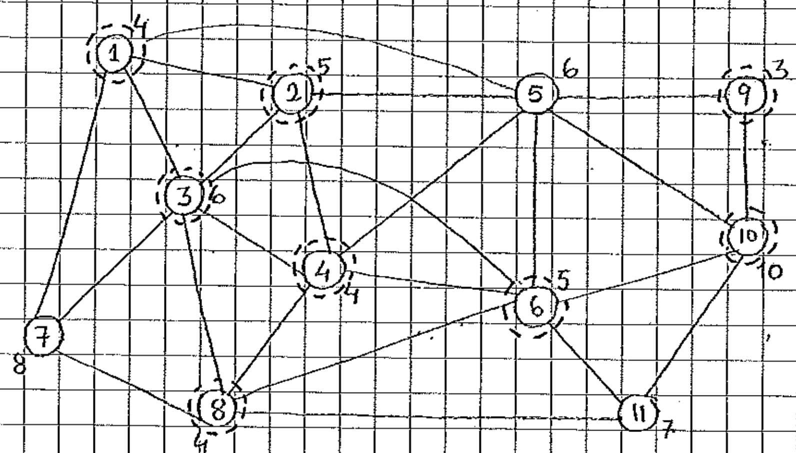
STEP 1: Sort the nodes in the ascending order of $f_i$
Pick the node $j^*$ at the top of the list.

$C = C \cup \{j^*\}$; erase all edges incident to $j^*$ from the graph

Recompute the $f_i$ (if the # of edges included to $i$ is zero set $f_i = +\infty$)

STEP 2: Check whether all edges have been covered. If so stop o/w go to Step 1.

EX



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\delta$: | 1 | 5/4 | 1 | 4/5* | 1 | 5/6 | 8/3 | 4/5* | 3/2 | 10/4 | 7/3 | $C=\{4\}$ |
| | 1* | 5/3 | 6/5 | | 6/5 | 1* | 8/3 | 1* | 3/2 | 10/4 | 7/3 | $C=\{4,1\}$ |
| | | 5/2 | 6/4 | | 6/4 | 1* | 4 | 1* | 3/2 | 10/4 | 7/3 | $C=\{4,1,6\}$ |
| | | 5/2 | 6/3 | | 6/3 | | 4 | 4/3* | 3/2 | 10/3 | 7/2 | $C=\{4,1,6,8\}$ |
| | | 5/2 | 3 | | 6/3 | | 8 | | 3/2* | 10/3 | 7 | $C=\{4,1,6,8,9\}$ |
| | | 5/2* | 3 | | 6/2 | | 8 | | | 10/2 | 7 | $C=\{4,1,6,8,9,2\}$ |
| | | | 6 | | 6 | | 8 | | | 5* | 7 | $C=\{4,1,6,8,9,2,10\}$ |
| | | | 6* | | $\infty$ | | 8 | | | | $\infty$ | $C=\{4,1,6,8,9,2,10,3\}$ |
| | | | | | $\infty$ | | $\infty$ | | | | $\infty$ | |

Rounding Heuristic for MECN: (RH) is a 2-approximation alg.

STEP 0: $C_{RH} \neq 0$, Solve LP Relaxation

$$y^* \rightarrow \left[ \begin{array}{ll} \text{Min } \sum_{i \in V} w_i y_i = Z^{LP} & \\ \text{St} & \\ \quad y_i + y_j \geq 1 & \forall (i,j) \in E \\ \quad 0 \leq y_i \leq 1 & \end{array} \right] \text{LP Relaxation}$$

STEP 1: For each $i \in V$ do

if $y_i^* \geq 1/2$ then $C_{RH} = C_{RH} \cup \{i\}$

STEP 2: Output $C_{RH}$ as a feasible cover.

Observation 1 = RH produces a cover.

Why? Imagine (suppose) $C_{RH}$ is not a cover, then $\exists (i,j) \in E$ for which neither $i \in C_{RH}$ nor $j \in C_{RH} \Rightarrow y_i^* < 1/2$

$y_j^* < 1/2$ IMPOSSIBLE

So $C_{RH}$ has to be cover.

GIPTA

Observation 2:- Let us call the value of $C_{RH}$, $Z_{RH} = \sum_{i \in C_{RH}} w_i$, Let us call the value of the optimal cover $Z^{OPT}$
Then,

$$\boxed{Z_{RH} \leq 2 \cdot Z^{OPT}}$$

Why? For any $i \in C$, $y_i^* \geq 1/2 \Rightarrow w_i^* y_i^* \geq 1/2 \, w_i^*$

$$\Rightarrow 2w_i^* y_i^* \geq w_i^* \quad \forall i \in C_{RH}$$

$$\therefore Z_{RH} \leq 2Z^{OPT} \Leftrightarrow Z_{RH} \sum_{i \in C_{RH}} w_i \leq 2 \sum_{i \in C_{RH}} w_i y_i^* \leq \underbrace{2 \sum_{i \in V} w_i y_i^*}_{2Z^{LP} \leq 2Z^{OPT}}$$

Another 2 Approximation Heuristic for MECN (DH)

(min cardinality version $w_i = 1$)

STEP 0: $C = \emptyset$

STEP 1: Select $(i,j) \in E$; $C = C \cup \{i,j\}$

Delete all edges incident to $i$ and to $j$ from the graph.

$E = E \setminus \{ \quad \}$

If uncovered edges remain, go back to 1 o/w STOP

$$\boxed{C_{DH} \leq 2Z^{OPT}}$$

# CHAPTER 4/ DYNAMIC PROGRAMMING

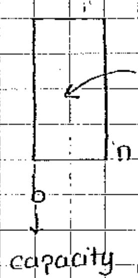
It is a way of formulating optimal decision problem by breaking the problem down to sequential stages: starting from the easiest problem, and synthesizing the solution to the entire problem in a sequential fashion.

Ingredients of DP:

1. Stage
2. State: description of the current stage of the system.
3. Control (decision variable)
4. Cost-to-go-function (measure to optimize)
5. Recursive functional
6. Boundary initial conditions.

## 4.1. FIRST ENCOUNTER: INTEGER KNAPSACK PROBLEM

There are items of various types
Thief breaks into a house:




- Jewelry
- Cash
- Compute equip.
- TV
- Video equip.
- Clothing
- Art

$$\text{Max} \sum_{J=1}^{n} C_J X_J$$

St

$$\sum a_J X_J \leq b$$

$$X_J \geq 0 \text{ integer}$$

1. Stage: the type of
2. State: available space in the sack
3. Control: amount (#) of item type $J$ to take in
4. $f_k(Y)$ = optimum profit attainable with items $1, \dots, k$ available budget equal to $y$.

Let us look $f_k(y)$ more closely:

$$f_k(y) = \max \sum_{J=1}^{k} C_J X_J \quad \text{St} \quad \sum_{J=1}^{k} a_J X_J \leq y, \quad X_J \geq 0 \text{ int. } J = 1, \dots, k$$

$$= \text{Max}_{X_k = 0,1,2,\dots [Y/a_k]} \left\{ C_k X_k + \left\{ \max \sum_{J=1}^{k-1} C_J X_J \right\} \right\} \quad \text{St} \quad \sum_{J=1}^{k-1} a_J X_J \leq y - a_k X_k, \quad \underbrace{X_J \geq 0 \text{ int } J = 1, \dots, k-1}_{f_{k-1}(y - a_k X_k)}$$

5. $$\boxed{f_k(Y) = \max_{X_k = 0,1,\dots,\lfloor y/a_k \rfloor} \{ C_k X_k + f_{k-1}(y - a_k X_k) \} \quad \forall k = 1, \dots, n}$$

6. $f_0(y) = 0 \; \forall y > 0$

EX: Max $11x_1 + 7x_2 + 5x_3$
St
$6x_1 + 4x_2 + 3x_3 \leq 15$ $x_1, x_2, x_3 \geq 0$ integer.

Initial Condition $f_0(y) = 0$ $y = 0, \ldots, 15$

Stage 1

| $y$ | $x_1$ | $f_1(y)$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 0 | 0 |
| 5 | 0 | 0 |
| 6 | 0,1 | max (0,11) = 11 |
| 7 | 0,1 | 11 |
| 8 | 0,1 | 11 |
| 9 | 0,1 | 11 |
| 10 | 0,1 | 11 |
| 11 | 0,1,2 | 11 |
| → 12 | 0,1,2 | max (0,11,22*) = 22 |
| 13 | 0,1,2 | 22 |
| 14 | 0,1,2 | 22 |
| 15 | 0,1,2 | 22 |

Stage 2

| $y$ | $x_2$ | $f_2(y)$ | |
|---|---|---|---|
| 0 | 0 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 0 | |
| 3 | 0 | 0 | |
| 4 | 0,1* | max {0, 7} | = 7 |
| 5 | 0,1 | max {0, 7} | = 7 |
| 6 | 0,1 | max {11, 7} | = 11 |
| 7 | 0,1 | max {11, 7} | = 11 |
| 8 | 0,1,2 | max {11, 7, 14} | = 14 |
| 9 | 0,1,2 | max {11, 7, 14} | = 14 |
| 10 | 0,1,2 | max {11, 18, 14} | = 18 |
| 11 | 0,1,2 | max {11, 18, 14} | = 18 |
| → 12 | 0*,1,2,3 | max {22*, 18, 14, 21} | = 22 |
| 13 | 0,1,2,3 | max {22, 18, 14, 21} | = 22 |
| 14 | 0,1,2,3 | max {22, 18, 25, 21} | = 25 |
| 15 | 0,1,2,3 | max {22, 18, 25, 21} | = 25 |

$f_2(y) = \max \{7x_2 + f_1(y - 4x_2)\}$

$x_2 = 0, 1, \ldots, \lfloor y/4 \rfloor$

Stage 3

| $y$ | $x_3$ | $f_3(y)$ |
|---|---|---|
| 15 | 0, 1*, 2, 3, 4, 5 | 25, 27*, 24, 26, 20, 25 |

$f_3(y) = \max \{5x_3 + f_2(y - 3x_3)\}$

$x_3 = 0, 1, \ldots, \lfloor y/3 \rfloor$

to $f_2(12)$

## 4.2 CAPITAL BUDGETING / RESOURCE ALLOCATION BY DP

A firm owns 3 plants and contemplates expansion projects for its operation. The management asked each plant to propose expansion projects. The plants submitted to the management projects with a figure for initial expenditure / investment $c(k)$ and projected revenue $r(k)$ for each level $k$ at which the project can be implemented.

| | PLANT 1 | | PLANT 2 | | PLANT 3 | | |
|---|---|---|---|---|---|---|---|
| LEVEL | c | r | c | r | c | r | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | (in M) |
| 2 | 1 | 5 | 2 | 8 | 1 | 4 | |
| 3 | 2 | 6 | 3 | 9 | | | |
| 4 | | | 4 | 12 | | | |

Find the most profitable assortments of project levels By DP

GIPTA

DP Methodology

1. Stages: Plant $\in \{1,2,3\}$
2. States: Budget available $\in \{0,1,2,3,4,5\}$, Control var = x
3. Cost to go function: $f_j(x)$ = max profit attainable for plants j to n with a budget of y.
4. Optimal Value thought as a function of cost-to-go function; $f_1(5)$
5. DP Recursion $f_3(y)$ $y = 0, \dots, 5$
6. Initial / Boundary Conditions

(Each project can be enacted at a single level. The total expansion budget is 5.)

budget ↘ level

| b | $X_3$ | $f_3(\cdot)$ |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 2 | 4 |
| 2 | 2 | 4 |
| 3 | 2 | 4 |
| 4 | 2 | 4 |
| 5 | 2 | 4 |

$f_2(y) = ?$

$$f_2(2) = \max\{\underbrace{0 + f_3(2)}_{X_2=1},\ \underbrace{8 + f_3(0)}_{X_2=2}\}$$

↳ I implement p.2 at level 2.

$$f_j(y) = \max\{r_j(X_j) + f_{j+1}(y - c_j(X_j))\}$$

$X_j \leq y$ pos integer $X_j = 0, 1, \dots y$

6. Boundary Cond. $f_3(y) = \max r_3(X_3)$ $X_3 = 0, 1, \dots, y$

budget ↘

| y | $X_2$ | $f_2$ |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 4 |
| 2 | 2 | 8 |
| 3 | 2 | 0+4, 8+4, 9+0 = 4, 12, 9 = 12 |
| 4 | 3 | 0+4, 8+4, 9+4, 12+0 = 13 |
| 5 | 4 | 0+4, 8+4, 9+4, 12+4 = 16 |

$f_1(5)$, I dont need to compute others.

$$f_1(5) = \max\{0 + f_2(5);\ 5 + f_2(4);\ 6 + f_2(3)\} = \boxed{18}$$

$5 + 13 = 18$; $6 + 12 = 18$

$\boxed{X_1^* = 2 \ (1m) \quad X_2^* = 3 \ (3m) \quad X_3^* = 2 \ (2m)}$

$\boxed{X_1^* = 3 \ (2M) \quad X_2^* = 2 \ (2M) \quad X_3^* = 2 \ (1m)}$

## 4.3 EQUIPMENT REPLACEMENT BY DP

A company needs to plan its Xerox machine policy for the next 5 years. A new machine costs a \$1000. The company can use a machine up to 3 years.

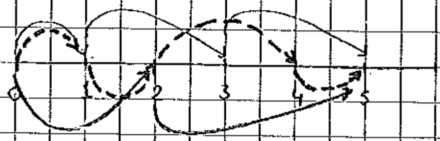
The machine has the salvage value equal to

800 after 1 years of use, (60 dollars for maint.)
600 2 years of use, (80 dollars for maint.)
500 after 3 years of use, (120 dollar for maint.)

Each year the machine is in use brings about maintenance costs, which are 60, 80 and \$120/year respectively. Assuming the company starts its 5 year horizon with a brand new machine, find the optimal replacement / maintenance policy for the horizon by DP.

1. Stages: Years
2. States: Age of the machine
3. Control: Keep / Replace (we dont have to quantify)
4. Cost-to-go function: Total Cost (Maintenance + buy new machine − salvage)

$f_t(x)$ = optimal cost of best policy from end of year $t$, to end of year $T$ ($T=5$) in our example starting with a machine $x$ years.




5. Optimal Value: $f_0(0) = ?$
6. DP Recursion:
7. Boundary / Initial Condt.:

$$f_4(1) = \min\{\overbrace{\underset{R}{1000+60-800-800}}^{-740};\ \underset{K}{80-600}\} = -740$$

$$f_4(2) = \min\{\underbrace{1000+60-600-800}_{460};\ \underbrace{120-500}_{-380}\} = -380$$

$$f_4(3) \underset{R}{=} 1000+60-500-800 = -240$$

$x = 1,2$ $\quad f_t(x) = \min\{\overset{R}{1000 - S_x + m_1 + f_{t+1}(1)},\ \overset{K}{m_{x+1} + f_{t+1}(x+1)}\}$ cost to go function

$x = 3$ $\quad f_t(3) = 1000 + m_1 - S_3 + f_{t+1}(1)$

$$f_3(1) = \min\{\overset{R}{\underbrace{1000+60-800+\overset{-740}{f_4(1)}}_{-340}};\ \overset{K}{80+\overset{-380}{f_4(2)}}\} = -480$$

$$f_3(2) = \min\{1000+60-600+f_4(1);\ 120+\overset{-240}{f_4(3)}\} = -280$$

$$f_3(3) = 1000 \underset{*}{-60} - 500 + \underset{-740}{f_4(1)} = -180$$

End of period 2;

$$f_2(1) = \min\{1060 - 800 + \overset{-480}{f_3(1)};\ \overset{*}{80 + \overset{-280}{f_3(2)}}\} = -200$$

$$f_2(2) = \min\{1060 - 600 + \overset{-480}{f_3(1)};\ \overset{*}{120 + \overset{280}{f_3(2)}}\} = -160$$

$$f_2(3) = 1060 - 500 + \underset{-480}{f_3(1)} = 80$$

End of period 1;

$$f_1(1) = \min\{1060 - 80 + \overset{-200}{f_2(1)};\ \overset{*}{80 + \overset{-160}{f_2(2)}}\} = -80$$

$$f_0(0) = 1000 + 60 + f_1(C_1) = +1080 \quad (?)$$

K K K R R (mark who is the winner!)

## 4.4 NON ADDITIVE RECURSION

The WHO is financing three independent research teams that are trying to develop a vacination. The project is considered a failure if none of the teams is able to produce a vaccination within a time frame. The teams prob of failure are 0.4, 0.6 and 0.8 respectively.

Therefore, the prob of failure of the project is $0.4 \times 0.6 \times 0.8 = 0.192$, which is deemed too high by the WHO. Hence, the WHO decides to allocate two prominent scientists to the project, to minimize the prob of failure if the impact of scientists on the teams are as follows;

| # of scientists | Team 1 | Team 2 | Team 3 |
|---|---|---|---|
| 0 | 0.4 | 0.6 | 0.8 |
| 1 | 0.2 | 0.4 | 0.5 |
| 2 | 0.15, $P_1(2)$ | 0.2 | 0.3 |
| | $P_i(X_i)$ | | |

Objective Function $\quad f_1(y) = \min \prod_{i=1}^{3} P_i(X_i)$

$$\sum_{i=1}^{3} X_i = y \qquad X_i \in \{0,1,2\}\ \forall i = 1,2,3$$

$f_j(y)$ = min prob failure for team j to 3 if scientists are allocated to these teams

cost to go function

Cont'd

$$f_j(y) = \min_{x_j \in \{0,\dots,y\}} p_j(x_j) \left\{ \begin{array}{l} \min \prod_{k=j+1}^{n} p(x_k) \\ x_{j+1},\dots \\ \text{st } \sum_{k=j+1}^{n} x_k = y - x_j \\ x_k \in \{0,1,\dots,y-x_j\} \end{array} \right\} f_{j+1}(y-x_j)$$

$$f_j(y) = \min \; p_j(x_j) f_{j+1}(y-x_j) \quad x_j \in \{0,1,\dots,y\}$$

$f_3(0) = 0.8$ $\quad$ $f_2(0) = 0.6 \times 0.8 = 0.48$

$f_3(1) = 0.5$ $\quad$ $f_2(1) = \min\{0.6 \times 0.5,\ 0.4 \times 0.8\} = 0.3$

$f_3(2) = 0.3$ $\quad$ $f_2(2) = \min\{0.6 \times 0.3,\ 0.4 \times 0.5,\ 0.2 \times 0.8\} = 0.16$

$$f_1(2) = \min\{\underset{0.064}{0.4 \times 0.16};\ \underset{0.06}{0.2 \times 0.3},\ \underset{0.072}{0.15 \times 0.48}\}$$

$\boxed{x_1^* = 1}$ $\quad$ $\boxed{x_2^* = 0}$ $\quad$ $\boxed{x_3^* = 1}$

**Question:** A Bilkent IE student is taking 3 courses. It is important that she does not fail all of them. If the probability of failing German is $p_2$ and the prob of failing IE342 is $p_3$, the probability of failing in Chinese is $p_1$. Failing all of them with probability of $p_1 p_2 p_3$. She has four hours left to study. How should she allocate these 4 hours among the three subjects so as to minimize the problem of failing all three courses? (By DP)

## 4.5 STOCHASTIC DP

Consider the following coin tossing game. a fair coin will be tossed 4 times. Before each toss you can wage 0, 1, 2 TL (if you have the means). You begin with 1 TL and want to maximize the probability of ending with 5 TL.

What is the optimal betting strategy?

min $k \leq \{x, 2\}$ $\quad$ $\frac{1}{2}$ $x - k$

win $\quad$ $\frac{1}{2}$ $x + k$

Cont'd

Stages: coin toss
State: Budget
Control: amount of bet

Cost-to-go function: $f_i(x)$ = max prob of ending with 5 TL starting from coin toss no $i$ up to coin toss no 4 starting with a budget of $x$.

$$\boxed{f_1(1) = ?}$$

Recursive DP Equation: $f_i(x) = \max_{k_i \leq \min\{2, x\}} \left\{ \tfrac{1}{2} f_{i+1}(x - k_i) + \tfrac{1}{2} f_{i+1}(x + k_i) \right\}$

Boundary Conditions: 0, 1, 2, 3, 4, 5
end of the Last Coin Toss

$f_5(5) = 1$
$f_5(0) = 0$

1 2 3 4

$f_4(0) = 0$ $\quad f_4(2) = 0$
$f_4(1) = 0$ $\quad f_4(3) = \max_{k_4 = 0,1,2} \{0, 0, \tfrac{1}{2} f_5(1) + \tfrac{1}{2} f_5(5)\} = \tfrac{1}{2}$

0 wage, 1 wage, win the bet.

$f_4(4) = \min \{0, \tfrac{1}{2} f_5(3) + \tfrac{1}{2} f_5(5); \tfrac{1}{2} f_5(2) + \tfrac{1}{2} f_5(6)\}$

$= \tfrac{1}{2}$ (Finish this!)

## 4.6 WAGNER-WHITIN INV MODEL

| $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ |
|---|---|---|---|---|---|
| 1 | | | | | T |

$I_0 = 0$ inventory position (end of period 0)

$$\left.\begin{aligned} I_1 &= I_0 + X_1 - d_1 \\ I_2 &= I_1 + X_2 - d_2 \\ &\;\vdots \\ I_T &= I_{T-1} + X_T - d_T \end{aligned}\right\} (\bullet)$$

$C_t(\cdot)$ = cost of p. one unit at $t$.

$h_t(\cdot)$ = cost of stocking one unit at time $t$.

$$\text{Min} \sum_{t=1}^{T} C_t(x_t) + \sum_{t=1}^{T} h_t(I_t)$$

St $(\bullet)$

$$\left.\begin{aligned} I_t &\geq 0 \\ X_t &\geq 0 \end{aligned}\right\} t = 1, \ldots, T.$$

DP

$v_t(I)$ = Optimum cost of the best production/inventory policy starting from begining period $t$ up to end of $T$ with an inventory position of $I$.

$$v_1(0) = ?$$

$$v_t(I) = \min_{X_t \in X_t} \left\{ C_t(X_t) + h_t(I + X_t - d_t) + v_{t+1}(I + X_t - d_t) \right\}$$

(curse of dimensionality)

$$v_t = \min \left\{ C_T(X_T) + \underbrace{v_{T+1}(0)}_{0} \right\}$$

Optimality Property $\exists$ an optimal solution $(X^*, I^*)$ such that $\forall\ t = 1, \ldots, T$

(FINAL) TRICK !

$$I^*_{t-1} X^*_t = 0$$

(There exist an optimal solution, all solutions maybe does not have that property. All optimal strategies.

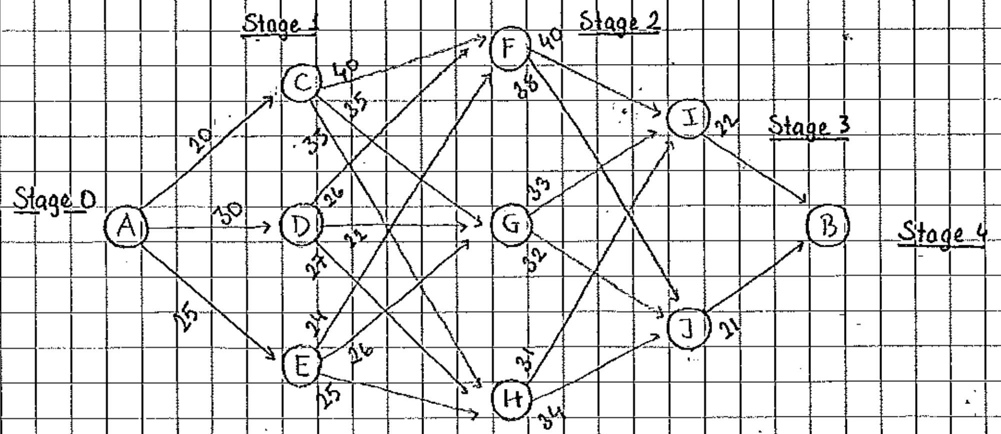
①  ②  ③  ④

$d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$

$I_0 = 0 \quad I_1 = d_2 + d \quad I_2 + d_3 I_2 > 0$

## 4.7 SHORTEST PATH COMPUTATIONS BY DP

A traveling salesperson has to go from A to B using the network sub. below. The number $c_{ij}$ on the arc $(i,j)$ represents the cost of our insurance policy from city $i$ to city $j$. Find by DP a route of smallest total insurance cost.



Stages: Obvious
States: the city where currently the d.p is
Control: the city go to next.
Cost-to-go function ($V_k(i)$) minimum total cost of the shortest path i at stage k to the destination (B).
Optimal Value ($V_0(A)$)
Backward Recursion: $V_4(B) = 0$ Boundary condition

for all stages $k$, and $i \in C_k$: $V_k(i) = \underset{j \in S(i)}{\min}^{\max} \{ c_{ij} + V_{k+1}(j) \}$

$j \in S(i)$ = cities that can be reached by one arc $(i,j)$ at the next stage.

$V_3(I) = 22 \quad V_3(J) = 21$

$V_2(F) = \min\{40+22, 38+21\} = 59$

$V_2(G) = \min\{33+22, 32+21\} = 53$

$V_2(H) = \min\{\underbrace{31+\overset{*}{2}2}_{53}, 34+21\} = 53$

$V_1(C) = \min\{40+59; 35+53; 33\overset{*}{+}53\} = 86$

$V_1(D) = \min\{28+59; 2\underset{*}{2}+53; 27+53\} = 75$

$V_1(E) = \min\{24+59; 26+53; 25\underset{*}{+}53\} = 78$

$V_0(A) = \{\underbrace{20+86}_{106}; \underbrace{30+75}_{105}; \underbrace{25+78}_{103}\} = 103$

A - E - H - I - B shortest route.

Forward Recursion:

$Z_k(i)$ = the cost of the shortest route from A to i at stage k

$Z_0(A) = 0 \quad Z_4(B) = ? , \quad Z_k(i) = \underset{j \in S^-(i)}{\min} \{ c_{ji} + Z_{k-1}(j) \}$

$S^-(i)$: set of nodes connecting from $C_{k-1}$ to i by an arc $(j,i)$.

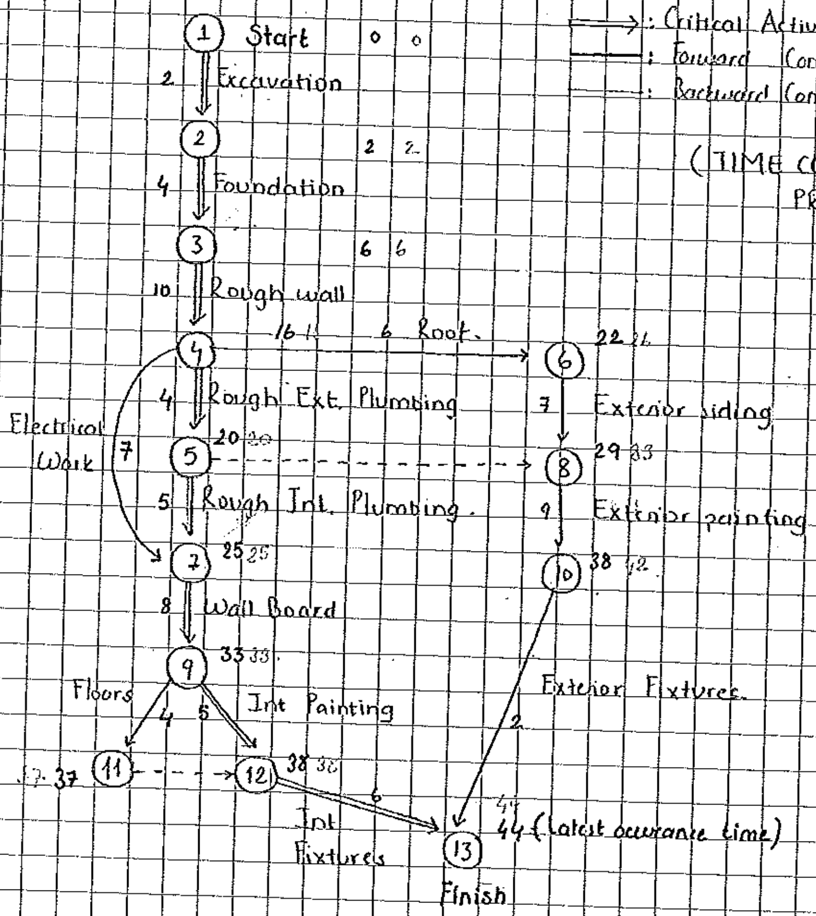
# CHAPTER 5 \ PROJECT SCHEDULING

The problem is to breakdown a project into various tasks with same procedure relationships among them and durations. How much time the project is going to take from start to completion?

Tasks Involved in constracting a house

| Procedure (Activities) | Procedure |
|---|---|
| 1. Excavation (Harfiyat) | — |
| 2. Foundation | 1 |
| 3. Rough Wall | 2 |
| 4. Rough Extrıor Plumbing | 3 |
| 5. Rough Interior Plumbing | 4 |
| 6. Electrical Work | 3 |
| 7. Roofing | 3 |
| 8. Exterior siding | 7 |
| 9. Exterior painting | 8 |
| 10. Wall board | 5,6 |
| 11. Floors | 10 |
| 12. Interior painting | 10 |
| 13. Exterior fixtures | 9 |
| 14. Interior fixtures | 11,10 |

Constructing a network




→ : Critical Activities.
— : Forward Comp.
— : Backward Comp.

(TIME COST TRADE OFF PROBLEM)

GIPTA

$E_i$ = Earliest occurance time of event $i$ (1)
$L_i$ = Latest occurance time of event $i$. (2)
$ES_{ij}$ = Earliest start time for activity $(i,j)$
$LS_{ij}$ = Latest start time for activity $(i,j)$ (Given)
$t_{ij}$ = duration of activity $(i,j)$
$EF_{ij}$ = Earliest finish time of activity $(i,j)$.
$LF_{ij}$ = Latest finish time of act $(i,j)$.
$S_{ij}$ = total slack in activity $(i,j)$ $(LF_{ij} - EF_{ij})$

(1) This is the earliest time an event can occur given that all proceding activities have been completed.

(2) This is the latest time an event can occur without delaying the project I completion time.

These activities, $(i,j)$ with zero slack are called "critical" activities. Any delay in a critical activity will increase the total project completion time.

FORward Computation of $E_i$, $ES_{ij}$, $EF_{ij}$

$$E_1 = 0$$

$$E_j = \max_{i:(i,j)\in PN} \{E_i + t_{ij}\}$$

$$\left.\begin{aligned} ES_{ij} &= E_i \\ EF_{ij} &= ES_{ij} + t_{ij} \end{aligned}\right\} \forall (i,j) \in A(PN) \text{ arc set of } PN$$

Backward Computation of $L_j$, $LS_{ij}$, $LF_{ij}$

$$L_n = E_n$$

$$L_j = \min_{(i,j)\in A(PN)} \{L_i - t_{ij}\}$$

$$LF_{ij} = \{L_j \,;\, LS_{ij} = LF_{ij} - t_{ij}\} \quad \forall (i,j) \in A(PN).$$

The critical activities constitute "the critical path".

∴ Hence the PM (Critical Path Method)

## 5.1 CRITICAL PATH METHOD

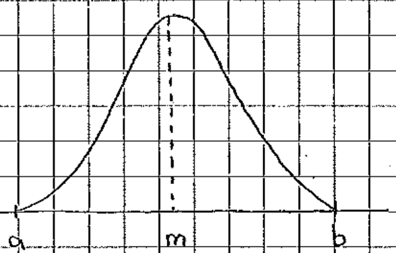
The critical path is the longest path in the project network. Therefore, the DP algorithms (forward and backward recursions) can be used to find the critical path.

## 5.2 PROJECT EVALUATION AND REVIEW TECHNIQUE (PERT)

a. Assume activity durations are stochastic and further assume that each activity has 3 estimates, m, a and b.

a ~ most optimistic estimate for the activity duration.
b ~ most pessimistic.
c ~ most likely estimate.

b. Model activity $(i,j)$ duration as a Beta distributed random variable.




$$\boxed{\sigma = \frac{b-a}{6}} \qquad \boxed{\bar{t} = \frac{1}{3}\left(2m + \tfrac{1}{2}(a+b)\right)}$$

mean of the random variable.

c. Assume that the project duration, which is the sum of all activity durations on the critical path is a normally distributed random variable with mean:

$$\boxed{\sum_{\substack{(i,j)\\ \text{critical}}} \bar{t}_{ij}} \ , \quad \boxed{\sigma^2 = \sum_{\substack{(i,j)\\ \text{critical}}} \sigma^2_{ij}}$$

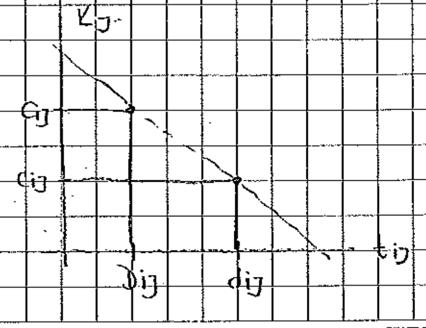
## 5.3 TIME COST TRADE OFF PROBLEM

You can accelerate certain activities by using more resources.

→ Shorter activity durations, so sooner project completition but bigger expenditure in terms of direct costs but you need to spend more.

Need an estimate of how much it costs to accelerate an activity?

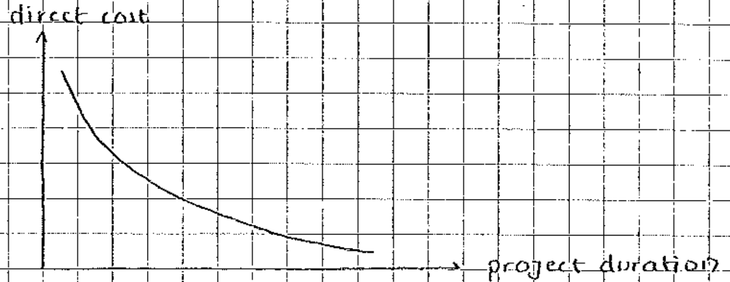
| | | |
|---|---|---|
| With existing equipment | $c_{ij}$ | $d_{ij}$ |
| With some more equipment | $C_{ij}$ | $D_{ij}$ |




GIPTA

Question: For a given maximum project completition time $\lambda$, how much direct cost shall we incur?

If I can solve this problem for fixed $\lambda \longrightarrow$ direct cost$(\lambda)$




The solution is by linear programming.
Let $t_{ij}$ be a non-negative variable for activity duration.

$$\text{Min} \sum_{(i,j) \in A(PN)} \left( K_{ij} - \rho_{ij} t_{ij} \right) \doteq \text{Max} \sum_{i,j} \rho_{ij} t_{ij}$$

St (1), (2), (3), (4), (5)

St.

$t_{ij} \leq d_{ij}$ (1)

$t_{ij} \geq D_{ij}$ (1)

$E_i + t_{ij} \leq E_j$ (2) $\forall (i,j) \in A(PN)$

critical path.

$E_1, \dots, E_n$ earliest occurances for events

project completition time

$E_n \leq \lambda$, $E_1 = 0$
(4) (3)

$E_i \geq 0$
(5)

Question: $E_j^* > E_i^* + t_{ij}^*$ for some $i$ in the opt question?