Part III — Decision

Based on lectures by Brian

Notes taken by Dexter Chua

Lent 2017-2018

Contents

1 Algorithms

1.1 Use an algorithm given in words

Example. The 'happy' algorithm is

- write down any integer
- square its digits and find the sum of the squares
- continue with this number 1
- repeat until either the answer is 1 (happy) or until you get trapped in a cycle (not happy).

Show that

(i) 70 is happy

(ii) 4 is unhappy

Proof. (i)

$$\begin{aligned} 7^2 + 0^2 &= 49 \\ 4^2 + 9^2 &= 96 \\ 9^2 + 7^2 &= 130 \\ 1^2 + 3^2 + 0^2 &= 10 \\ 1^2 + 0^2 &= 1 \end{aligned}$$

so 70 is happy

(ii)

$$\begin{aligned} 4^2 &= 16 \\ 1^2 + 6^2 &= 37 \\ 3^2 + 7^2 &= 58 \\ 5^2 + 8^2 &= 89 \\ 1^2 + 4^2 + 5^2 &= 42 \\ 4^2 + 2^2 &= 20 \\ 2^2 + 0^2 &= 4 \\ 4^2 &= 16 \end{aligned}$$

so 4 is unhappy

□

Example. Implement this algorithm

- Let $n + 1, A = 1, B = 1$
- Write down A and B
- Let $C = A + B$

- Write down C
- Let $n = n + 1, A = B, B = C$
- If $n < 5$ go to 3
- If $n = 5$ stop

Proof.

Instruction step	n	A	B	C	Write down
1	1	1	1		
2					1,1
3				2	
4					2
5	2	1	2		
6	Go to step 3				
3				3	
4					3
5	3	2	3		
6	Go to Step 3				
3				5	
4					5
5	4	3	5		
6	Go to Step 3				
3				8	
4					8
5	5	5	8		
6	Continue to Step 7				
7	Stop				

□

Example. The algorithm multiplies the two numbers A and B.

- Make a table with two columns
  Write A in the top row of the left hand column and B in the top row of the right hand column
- In the next row of the table write:
  - in the left hand column, the number that is half A, ignoring remainders
  - in the right hand column the number that is double B
- Repeat step 2 until you reach the row which has a 1 in the left hand column
- Delete all rows where the number in the left hand column is even
- Find the sum of the non-deleted numbers in the right hand column
  This is the product AB

Implement this algorithm when

(i) $A = 29$ and $B = 34$

(ii) $A = 66$ and $B = 56$

Proof. (i)

A	B	Step 4
14	68	Delete
7	136	
3	272	
1	544	
Total	986	

So $29 \times 34 = 986$

(ii)

A	B	Step 4
66	56	Delete
33	112	
16	224	Delete
8	448	Delete
4	896	Delete
2	1792	Delete
1	3584	
Total	3584	

So $66 \times 56 = 3696$

□

1.2 Implement an algorithm given in the form of a flow chat

There are three shapes of boxes which are used in the examination

Example. (i) Implement this algorithm using a trace table

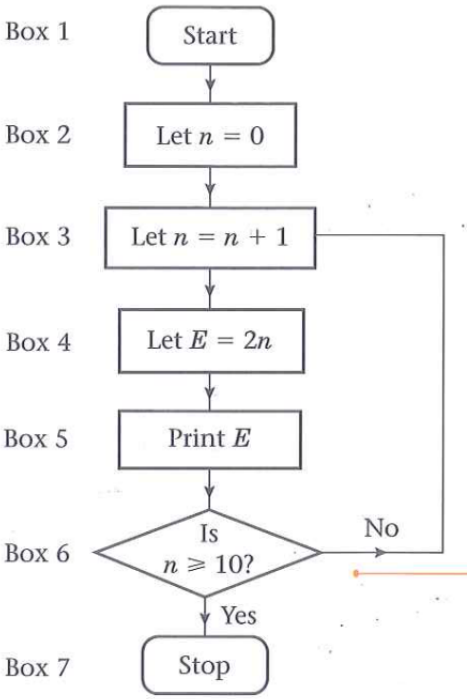
(ii) Alter box 4 to read 'Let $E = 3n$' and implement the algorithm again. How does this alter the algorithm?

Proof. (i)

n	E	box 6
14	68	no
7	136	no
3	272	no
1	544	no
Total	986	no

(ii)

n	E	box 6
66	56	no
33	112	no
16	224	no
8	448	no
4	896	no
2	1792	no
1	3584	no
Total	3584	no



□

1.3 Bubble sort

1.4 Binary search

1.5 Implement the three bin packing algorithms

2 Graphs and networks

2.1 Basic Graph theory

Definition (Vertices and Edges of Graph)**.** In the graph G, above

- the vertices (or nodes) are:
- the edges (or arcs) are:

Definition (Subgraph)**.** A subgraph of G is a graph , each of whose vertices belongs to G and each of whose edges belongs to G. It is simply a part of the original graph.

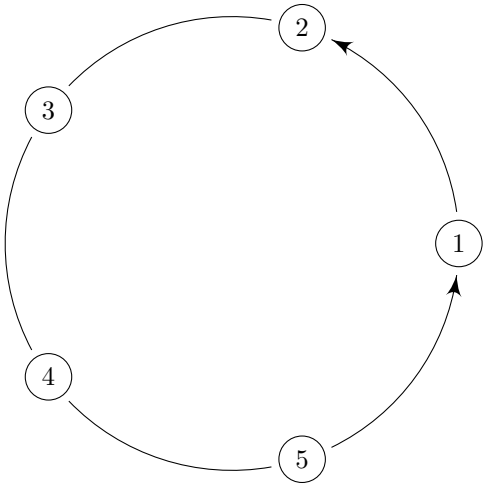
Definition (Degree)**.** The degree or valency or order of a vertex is the number of edges incident to it

Proposition. If the degree of a vertex is even , we say it has even degree

Definition (Path)**.** A path is a finite sequence of edges, such that the end vertex of one edge in the sequence is the start vertex of the next, and in which no vertex appears more than once.

Definition (Walk)**.** A walk is a path in which you are permitted to return to vertices more than once.

Definition (Cycle)**.** A cycle is a closed 'path'.



Definition (Connected)**.** Two vertices are connected if there is a path between them. A graph is connected if all its vertices are connected.

Definition (Loop)**.** A loop is an edge that starts and finishes at the same vertex

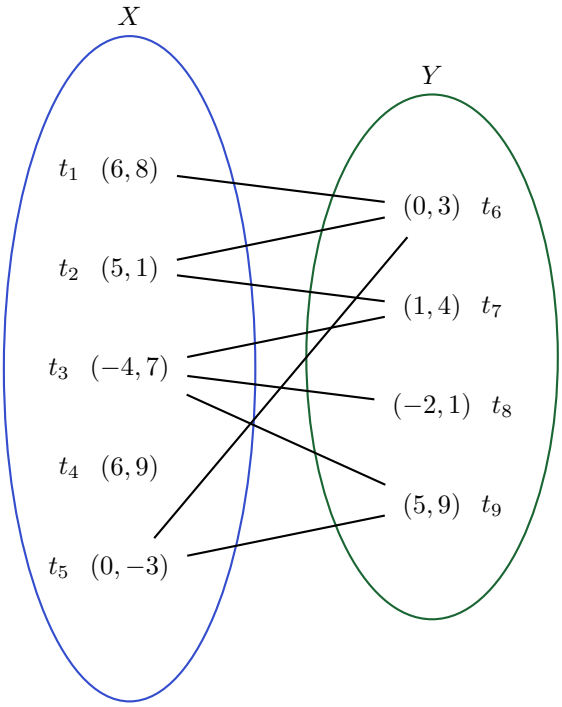
Definition (Simple graph)**.** A simple graph is one in which there are no loops and not have more than one edge connecting any pair of vertices

Definition (Digraph)**.** If the edges of a graph have a direction associated with them they are known as directed edges and the graph is known as a digraph.

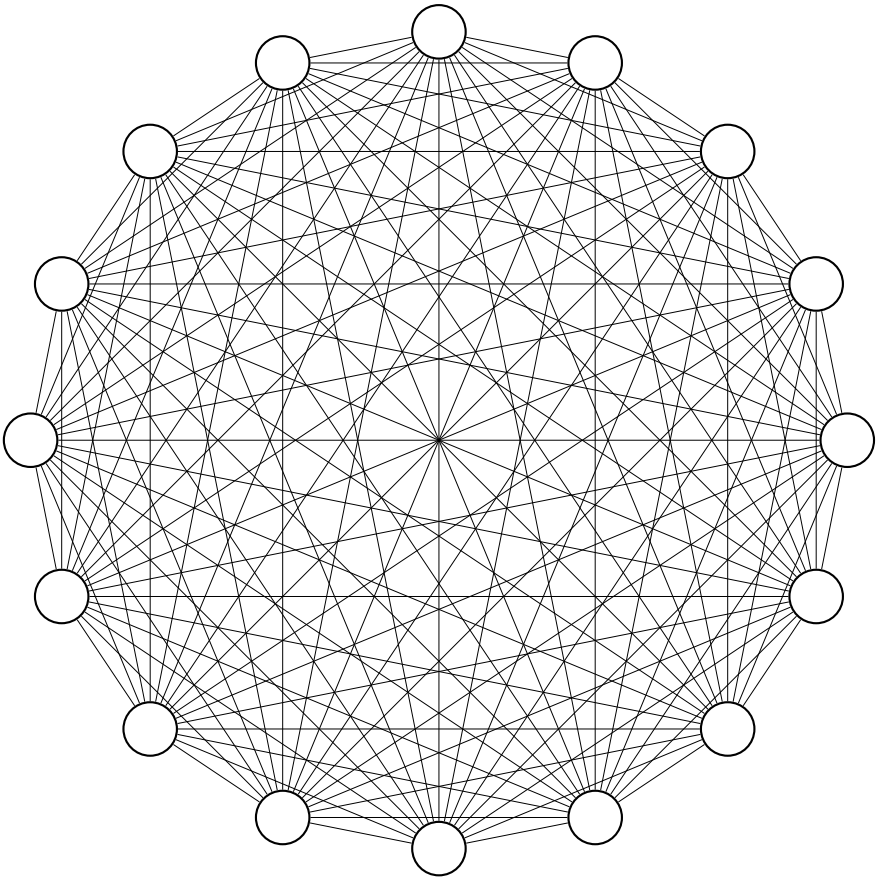
Definition (Tree)**.** A tree is a connected graph with no cycles

Definition (Spanning tree)**.** A spanning tree of a graph ,G is a subgraph which includes all the vertices of G and is also a tree.

Definition (Bipartite graph)**.** A bipartite graph consists of two sets of vertices , X and Y. The edges only join vertices in X to vertices in Y, not vertices within a set



Definition (Complete graph)**.** A complete graph is a graph in which every vertex is directly connected by an edge to each of the other vertices. If the graph has n vertices the connected graph is denoted by K_n.



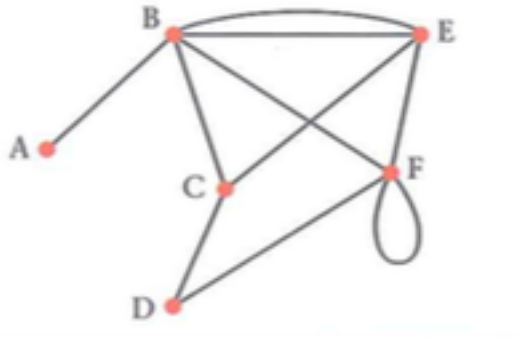
Definition (Complete bipartite graph)**.** A complete bipartite graph (denoted by $K_{r,s}$) in which there are r vertices in set X and s vertices in set Y.

Definition (Isomorphic graphs)**.** Isomorphic graphs are graphs that who the same information but are drawn differently

2.2 Adjacency Matrix and Distance matrix

Definition (Adjacency Matrix)**.** A adjacency matrix records the number of direct links between vertices

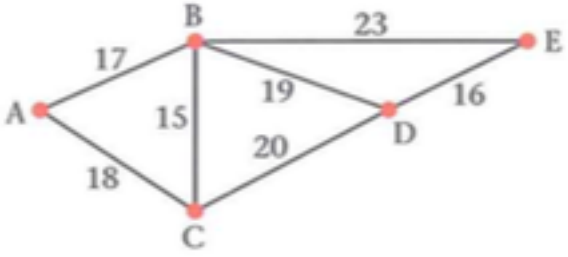
Example. Use an adjacency matrix to represent this graph



	A	B	C	D	E	F
A	0	1	0	0	0	0
B	1	0	1	0	2	1
C	0	1	0	1	1	0
D	0	0	1	0	0	1
E	0	2	1	0	0	1
F	0	1	0	1	1	2

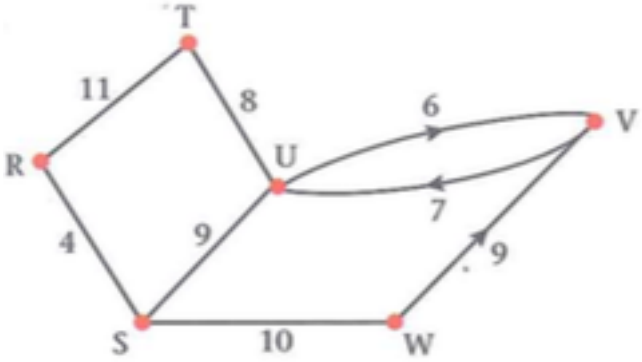
Definition (Distance Matrix)**.** A distance matrix records the weights on the edges. Where there is no edge , we write '−'

Example. Use a distance matrix to represent this network.



	A	B	C	D	E	F
A	0	1	0	0	0	0
B	1	0	1	0	2	1
C	0	1	0	1	1	0
D	0	0	1	0	0	1
E	0	2	1	0	0	1
F	0	1	0	1	1	2

Example. Use a distance matrix to represent this directed network.



	A	B	C	D	E
A	-	17	18	-	-
B	17	-	15	19	23
C	18	15	-	20	-
D	-	19	20	-	16
E	-	23	-	16	-

3 Algorithms on networks

3.1 Kruskal's algorithm

Kruskal's algorithm finds the shortest, cheapest or fastest way of linking all the nodes into one system
You can use Kruskal's algorithm to find a minimum spanning tree.

Definition (Minimum spanning tree)**.** A minimum spanning tree (MST) is spanning tree such that the total length of its edges is as small as possible. (An MST is sometimes called a minimum connectors).

Definition (Kruskal's algorithm)**.** Here is Kruskal's algorithm.

(i) Sort all the edges into ascending order of weight.

(ii) Select the edges of least weight to start the tree

(iii) Consider the next edge of least weight.

- If it would form a cycle with the edges already selected, reject it.
- If it does not form a cycle, add it to the tree.

If there is choice of equal edges, consider each in turn.

(iv) Repeat step 3 until all vertices are connected.

Example. Use Kruskal's algorithm to find a minimum spanning tree

Example.

3.2 Prim's algorithm

Like Kruskal's algorithm, Prim's algorithm finds the minimum spanning tree, but it uses a different approach.

Definition (Prim's algorithm)**.** Here is Prim's algorithm

(i) Choose any vertex to start the tree.

(ii)
- Select an edges of least weight that joins a vertex that is already in the tree to a vertex that is not yet in the tree.
- If there is a choice of edges of equal weight , choose randomly.

(iii) Repeat Step 2 until all the vertices are connected.

Example.

You can apply Prim's algorithm to a distance matrix

Definition (Prim's algorithm to a distance matrix)**.** Here is the distance matrix form of Prim's algorithm

(i) Choose any vertex to start the tree.

(ii) Delete the row in the matrix for the chosen vertex.

(iii) Number the column in the matrix for the chosen vertex

(iv) Put a ring round the lowest undeleted entry in the numbered columns. (If there is an equal choice, choose randomly.)

(v) The ringed entry becomes the nest edge to be added to the tree.

(vi) Repeats steps 2,3,4 and 5 until all rows are deleted.

Example.

3.3 Dijkstra's algorithm

Dijkstra's algorithm is used to find the shortest, cheapest or quickest route between two vertices.

Definition (Dijkstra's algorithm)**.** Here is Dijkstra's algorithm
(to find the shortest path form S to T through a network).

(i) Label the start vertex , S , with the final label ,0.

(ii) Record a working value at every vertex , Y, that is directly connected to the vertex , X , that has just received its final label.

- Working value at Y = final value at $X+$ weight of arc XY
- If there is already a working value at Y, it is only replaced if the new value is smaller.
- Ince a vertex has a final label it is not revisited and its working values are no longer considered.

(iii) Look at the working values at all vertices without final labels. Select the smallest working value. This now becomes the final label at that vertex. (If two vertices have the same smallest working value either may be given its final label first.)

(iv) Repeat steps 2 and 3 until the destination vertex , T, receives its final label.

(v) To find the shortest path, trace back from T to S. Given that B already lies on the route, include arc AB whenever final label of B - final label of A = weight of arc AB.

Example.

Example.

Example.

4 Route inspection (Chinese postman problem)

4.1 Traversable

Definition (Eulerian)**.** If all the degrees in a graph are even, then the graph is Eulerian. If precisely two degrees are odd, and all the rest are even, then the graph is semi-Eulerian

Definition (Traversable)**.** A graph is traversable if it is possible to traverse (travel along) every arc just once without taking your pen from the paper.

Proposition. A graph traversable if all the degrees are even

Proposition. A graph is semi-traversable if it has precisely two odd degrees. In this case the start point and the finish point will be the two vertices with odd degrees.

Proposition. A graph is not traversable if it has more than two odd degrees.

Example.

Example.

Example.

4.2 Chinese postman algorithm (route inspection algorithm)

You can use this algorithm to find the shortest route in a network that traverses every arc at least once and returns to the starting point.

Proposition. If all the vertices have even degree the network is traversable. The length of the shortest route will be equal to the weight of the network

Example.

Proposition.

Example.

Example.

Definition (Route inspection algorithm)**.** Here is the route inspection algorithm

(i) Identify any vertices with odd degree.

(ii) Consider all possible complete pairings of these vertices.

(iii) Select the complete pairing that has the least sum.

(iv) Add a repeat of the arcs indicated by this pairing to the network.

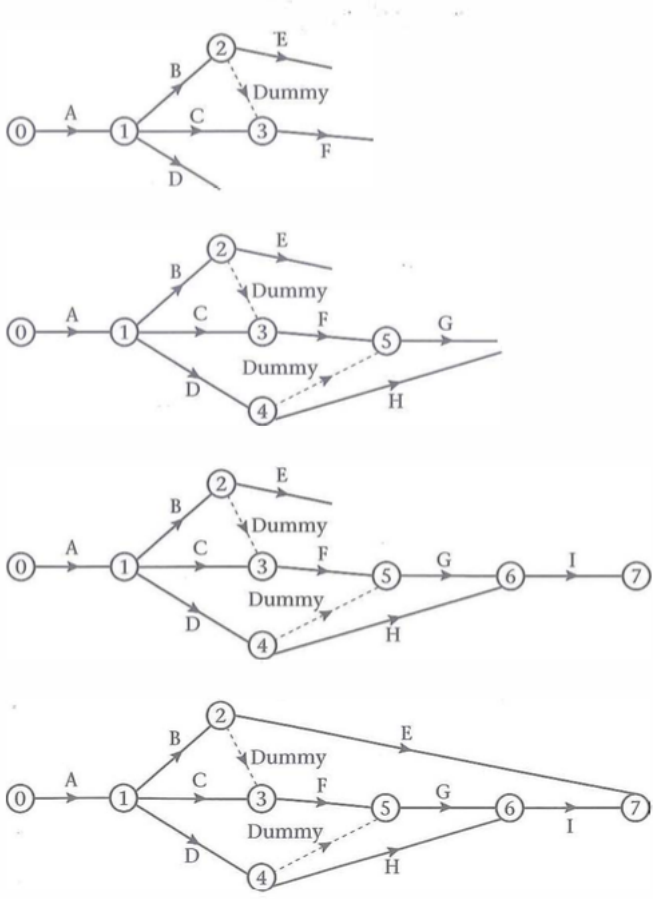
Example.

5 Critical path analysis

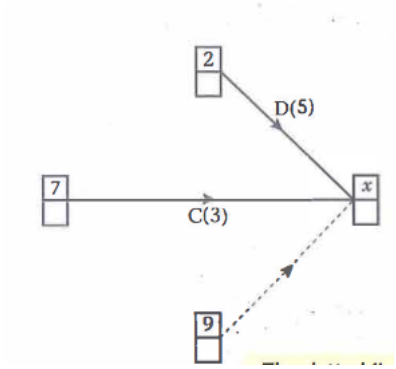
5.1 Basic calculation

Example. Draw an activity network using activity on arc for this precedence table. Use exactly two dummies.

Activity	Immediately preceding activities
A	-
B	A
C	A
D	A
E	B
F	B,C
G	D,F
H	D
I	G,H



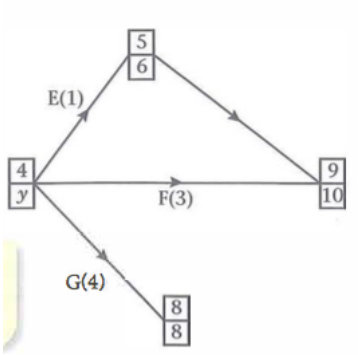
Example. The diagram shows part of an activity network. Calculate the value of x.



$$2 + 5 = 7$$
$$7 + 3 = 10$$
$$9 + 0 = 9$$

The largest is 10, so $x = 10$

Example. The diagram shows part of an activity network. Calculate the value of y.



$$6 - 1 = 5$$
$$10 - 3 = 7$$
$$8 - 4 = 4$$

The smallest is 4, so $y = 4$

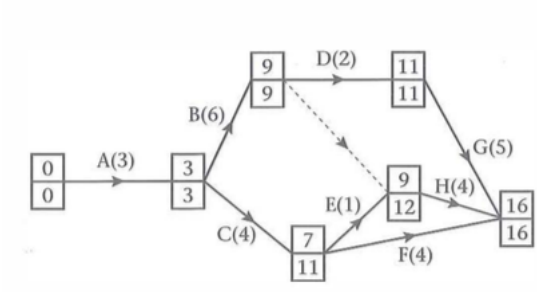
Definition (Critical activity and path)**.** The path(s) and node(s) which are directed to the results

5.2 Total float

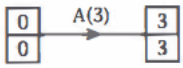
Definition (Total float)**.** The total float of an activity is the amount of time that its start may be delayed without affecting the duration of the project.

$$\text{total float} = \text{latest finish time} - \text{duration} - \text{earliest start time}$$

Example. Determine the total float of each activity in this activity network.

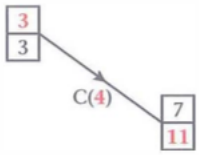


Activity A

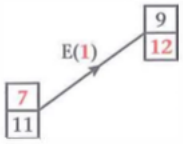


Activities B, D and G

Activity C

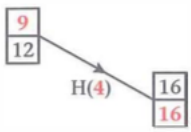


Activity E



Activity F

Activity H



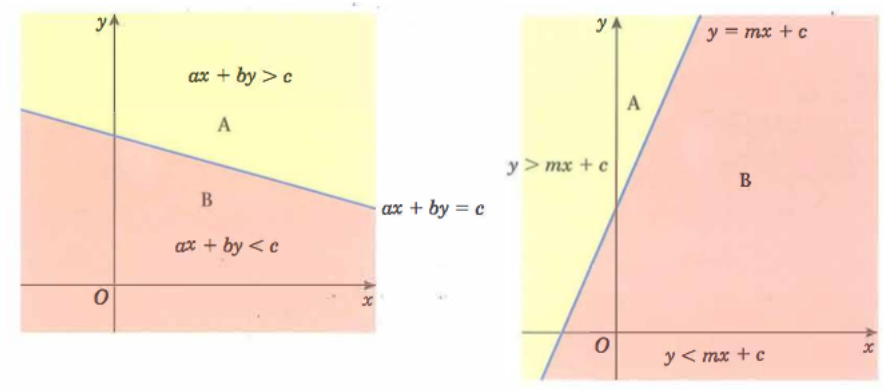
At Activity A, Any delay in the start time will affect the duration of the project. Total float = 0.

C : $11 - 4 - 3 = 4$
E : $12 - 1 - 7 = 4$
F : $16 - 4 - 7 = 5$
H : $16 - 4 - 9 = 3$

Example. Examples 11-14

5.3 Cascade (Gantt) charts

6 Linear programming



Remark. Using dotted line if the inequalities are $>$ or $<$

Example. Example 5-13

7 Matchings

Example. Example 3-6

8 Transportation problems

8.1 North-west corner method

Example. Use the north-west corner method to find an initial solution to the problem described in that example and shown in the table

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	180	110	130	290	14
Supplier B	190	250	150	280	16
Supplier C	240	270	190	120	20
Demand	11	15	14	10	50

Proof. Set up the table

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A					14
Supplier B					16
Supplier C					20
Demand	11	15	14	10	50

Look at the column demand first which is lowest, write down the value on left top corner

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11				14
Supplier B					16
Supplier C					20
Demand	11	15	14	10	50

write down the different between the row and you previous number on right

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11	3			14
Supplier B					16
Supplier C					20
Demand	11	15	14	10	50

write down the different between the column and you previous number on below

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11	3			14
Supplier B		12			16
Supplier C					20
Demand	11	15	14	10	50

write down the different between the row and you previous number on right

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11	3			14
Supplier B		12	4		16
Supplier C					20
Demand	11	15	14	10	50

write down the different between the column and you previous number on below

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11	3			14
Supplier B		12	4		16
Supplier C			10		20
Demand	11	15	14	10	50

write down the different between the row and you previous number on right

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11	3			14
Supplier B		12	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

This is the final table. All of the stock has been used and all of the demands met.

The number of occupied cells (routes used) in the table = number of supply points + number of demand points − 1 .
In this case the
number of occupied cells (routes used) = 6,
number of supply points = 3,
number of demand points = 4
and $6 = 3 + 4 - 1$.

Use this table, together with the table showing costs, to work out the total cost of the solution.

	W	X	Y	Z	Stock
A	180	110	130	290	14
B	190	250	150	280	16
C	240	270	190	120	20
Demand	11	15	14	10	50

The total cost of this solution is $(11 \times 180) + (3 \times 110) + (12 \times 250) + (4 \times 150) + (10 \times 190) + (10 \times 120) = 9010$ □

Definition (balanced)**.** When the total supply > total demand , we say the problem is unbalanced, If the problem is unbalanced we simply add a dummy demand point with a demand chosen so that total supply = total demand, with transportation costs of zero.

Example. Three outlets A, B and C are supplied by three suppliers X, Y and Z. The table shows the cost, in pounds, of transporting each unit, the number of units required at each outlet and the number of units available at each supplier.

	A	B	C	Supply
X	9	11	10	40
Y	10	8	12	60
Z	12	7	8	50
Demand	50	40	30	

Add a dummy demand point and appropriate costs to the table. and use the north-west corner method to obtain an initial solution.

Proof. Here you are

	A	B	C	D	Supply
X	9	11	10	0	40
Y	10	8	12	0	60
Z	12	7	8	0	50
Demand	50	40	30	30	150

	A	B	C	D	Supply
X	40				40
Y	10	40	10		60
Z			20	30	50
Demand	50	40	30	30	150

□

Definition (degenerate)**.** In a feasible solution to a transportation problem with m rows and n columns, if the number of cells used is less than $n + m - 1$, then the solution is degenerate

Example. (i) Demonstrate that the north-west corner method gives a degenerate solution and explain why it is degenerate

(ii) Adapt your solution to give a non-degenerate initial solution and state its cost.

	A	B	C	Supply
W	10	11	6	30
X	4	5	9	20
Y	3	8	7	35
Z	11	10	9	35
Demand	30	40	50	120

Proof. (i)

	A	B	C	Supply
W	30			30
X		20		20
Y		20	15	35
Z			35	35
Demand	30	40	50	120

This solution is degenerate since it fulfils all the supply and demand needs but only uses 5 cells WA, XB, YB, YC and ZC. There are 4 rows and 3 columns so a non-degenerate solution will use $4 + 3 - 1 = 6$ cells.

(ii)
Start by placing the largest possible number in the north-west corner.

	A	B	C	Supply
W	30			30
X				20
Y				35
Z				35
Demand	30	40	50	120

There are two possible initial solutions, depending on where you chose to place the zero

either

	A	B	C	Supply
W	30			30
X	0	20		20
Y		20	15	35
Z			35	35
Demand	30	40	50	120

Or

	A	B	C	Supply
W	30			30
X	0	20		20
Y		20	15	35
Z			35	35
Demand	30	40	50	120

□

8.2 Shadow costs

Definition. The sum of shadow costs of column and row equation to that overlaping number box

Example. Calculate the shadow costs given by the initial solution.

	W	X	Y	Z	Stock
A	180	110	130	290	14
B	190	250	150	280	16
C	240	270	190	120	20
Demand	11	15	14	10	50

Proof. Initial solution was

	W	X	Y	Z	Stock
A	11	3			14
B		12	4		16
C			10	10	20
Demand	11	15	14	10	50

Focus on the costs of the routes being used - the non-empty squares

	W	X	Y	Z	Stock
A	180	110			14
B		250	150		16
C			190	120	20
Demand	11	15	14	10	50

Putting $S(A) = 0$, from row 1 we get $D(W) = 180$ and $D(X) = 110$
the solve the equations
$S(A) + D(W) = 180$
$S(A) + D(X) = 150$

Shadow costs		180	110			
		Depot W	Depot X	Depot Y	Depot Z	Stock
0	Supplier A	180	110			14
	Supplier B		250	150		16
	Supplier C			190	120	20
	Demand	11	15	14	10	50

Now move to Row 2.
You know that $D(X) = 110$, so you find $S(B) = 140$
hence $D(Y) = 10$

Shadow costs		180	110	10		
		Depot W	Depot X	Depot Y	Depot Z	Stock
0	Supplier A	180	110			14
140	Supplier B		250	150		16
	Supplier C			190	120	20
	Demand	11	15	14	10	50

Move to Row 3
You know that $D(Y) = 10$, so we find
$S(C) = 180$ and hence that $D(Z) = -60$

Shadow costs		180	110	10	-60	
		Depot W	Depot X	Depot Y	Depot Z	Stock
0	Supplier A	180	110			14
140	Supplier B		250	150		16
160	Supplier C			190	120	20
	Demand	11	15	14	10	50

Done. □

Example. Calculate the shadow costs given by the initial solution.

	A	B	C	Supply
X	9	11	10	40
Y	10	8	12	60
Z	12	7	8	50
Demand	50	40	30	

Add a dummy demand point and appropriate costs to the table. and use the north-west corner method to obtain an initial solution.

Proof. Here you are

	A	B	C	D	Supply
X	9	11	10	0	40
Y	10	8	12	0	60
Z	12	7	8	0	50
Demand	50	40	30	30	150

	A	B	C	D	Supply
X	40				40
Y	10	40	10		60
Z			20	30	50
Demand	50	40	30	30	150

Shadow costs		9	7	11	3	
		A	B	C	D	Supply
0	X	40				40
1	Y	10	40	10		60
-3	Z			20	30	50
	Demand	50	40	30	30	150

□

8.3 Improvement indices

Definition (Improvement index)**.** The improvement index in sending a unit from a source P to a demand point Q is found by subtracting the source cost $S(P)$ and destination cost $D(Q)$ from the stated cost of transporting one unit along that route $C(PQ)$. i.e.

$$\text{Improvement index for PQ} = I_{PQ} = C(PQ) - S(P) - D(Q)$$

Definition (Optimal)**.** If there are no negative improvement indices.

Example. Calculate the improvement indices

Shadow costs		180	110	10	-60	
		Depot W	Depot X	Depot Y	Depot Z	Stock
0	Supplier A	180	110			14
140	Supplier B		250	150		16
160	Supplier C			190	120	20
	Demand	11	15	14	10	50

$BW = -130$
$CW = -120$
$CX = -20$
$AY = 120$
$AZ = 350$
$BZ = 200$

8.4 Stepping-stone method

Example. Obtain an improved solution and find the improved cost.

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	180	110	130	290	14
Supplier B	190	250	150	280	16
Supplier C	240	270	190	120	20
Demand	11	15	14	10	50

and the initial solution was

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11	3			14
Supplier B		12	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

at a cost of 9010

Proof. Use BW as the entering cell, since this gave the most negative improvement index, -130. So BW will be and increasing cell. We enter a value of θ into this cell

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11	3			14
Supplier B	θ	12	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

In order to keep the demand at W correct. you must therefore decrease the entry at AW, so AW will be a decreasing cell

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	$11-\theta$	3			14
Supplier B	θ	12	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

In order to keep the demand at A correct. you must therefore decrease the entry at AX, so AW will be a increasing cell

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	$11-\theta$	$3+\theta$			14
Supplier B	θ	12	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

In order to keep the demand at X correct. you must therefore decrease the entry at BX, so BX will be a decreasing cell

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	$11-\theta$	$3+\theta$			14
Supplier B	θ	$12-\theta$	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

Now choose a value for θ, the greatest value you can, without intoducing negative entries into the table. Look at the decreasing cells and see that the greatest value of θ is 11 (since $11 - 11 = 0$)

Replace θ by 11 in the table

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A	11−11	3+11			14
Supplier B	11	12−11	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

This gives the improved solution:

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A		14			14
Supplier B	11	1	4		16
Supplier C			10	10	20
Demand	11	15	14	10	50

this solution has a cost of 7580

As a double check, it is always true that

$$\text{New cost} = \text{cost of former solution} + \text{improvement index} \times \theta$$

In this case $7580 = 9010 + (-130 \times 11)$

You will notice that AW has become empty that is exiting cell □

Example. We also find an optimal solution

Proof. Second iteration
The improvement indices are:
$AW = 130$
$CW = 10$
$CX = -20$
$AY = 120$
$AZ = 350$
$BZ = 200$

So the new entering cell is CX, since this has the most negative improvement index

Applying the stepping-stone method gives

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A		14			14
Supplier B	11	$1-\theta$	$4+\theta$		16
Supplier C		θ	$10-\theta$	10	20
Demand	11	15	14	10	50

Looking at cells BX and CY we see that the greatest value for θ is 1

The new exiting cell will be BX, $\theta = 1$ and we get

	Depot W	Depot X	Depot Y	Depot Z	Stock
Supplier A		14			14
Supplier B	11		5		16
Supplier C		1	9	10	20
Demand	11	15	14	10	50

The new cost is 7560

Checking $7580 + (-20) \times 1 = 7560$

Third iteration

Shadow costs		70	110	30	-40	
		W	X	Y	Z	Stock
0	A	180	110	130	290	14
120	B	190	250	150	180	16
160	C	240	270	190	120	20
	Demand	11	15	14	10	50

There are no negative improvement indices so this solution is optimal (by Calculate the improvement indices) □

8.5 Transportation problem and linear programming

Example. Formulate the transportation problem as a linear programming problem. You must state your decision variables, objective and constraints.

	R	S	T	Supply
A	3	3	2	25
B	4	2	3	40
C	3	4	3	31
Demand	30	30	36	

Proof. Let x_{ij} be the number of units transported from i to j where $i \in \{A, B, C\}$ and $J \in \{R, S, T\}$ and $x_{ij} \geq 0$

$$\begin{aligned} \text{Minimise} C = 3x_{11} + 3x_{12} + 2x_{13} \\ + 4x_{21} + 2x_{22} + 3x23 \\ + 3x_{31} + 4x_{32} + 3x_{33} \end{aligned}$$

Subject to:

$$\begin{aligned} x_{11} + x_{12} + x_{13} \leq 25 \\ x_{21} + x_{22} + x_{23} \leq 40 \\ x_{31} + x_{32} + x_{33} \leq 31 \\ x_{11} + x_{21} + x_{31} \leq 30 \\ x_{12} + x_{22} + x_{32} \leq 30 \\ x_{13} + x_{23} + x_{33} \leq 36 \end{aligned}$$

□

9 Allocation (assignment) problems

9.1 Reduce cost matrics

To subtract the least value for each element of that row and then column repeatlly.

Example. Reduce the cost matrix

	Task W	Task X	Task Y	Task Z
Kris	12	23	15	40
Laura	14	21	17	20
Sam	13	22	20	30
Steve	14	24	13	10

Proof. reduce the least number of the row of each row

	Task W	Task X	Task Y	Task Z
Kris	0	11	3	28
Laura	0	7	3	6
Sam	0	9	7	17
Steve	4	14	3	0

reduce the least number of the column of each column

	Task W	Task X	Task Y	Task Z
Kris	0	4	0	28
Laura	0	0	0	6
Sam	0	2	4	17
Steve	4	7	0	0

If we can find a matching using only the cells showing a zero cost we will have found an optimal solution. In this case we can.

Oue optimal solution is

Kris does task Y Laura does task X Sam does task W and Steve does task Z

If you look at the original cost matrix the total cost is $15 + 21 + 13 + 10 = 59$

Check total cost was 59 which is also the value of all the row and column reductions we made $(12 + 14 + 13 + 10 + 7 + 3 = 59)$

□

9.2 Hungarian algorithm

(i)

(ii)

(iii)

(iv)

(v)

(vi)

(vii)

Example. 2-9

10 The travelling salesman problem

Example. example 2-10

11 Further linear programming

Example. 1-13

12 Game theory

Example. 1-12

13 Network flows

Example. 1-16

14 Dynamic programming

Example. 1-10