Nankai University MBA中心

Modern Logistics & Supply Chain Management

ML & SCM

Supply Networks
Minimal Spanning Tree

Dr. Wolfgang Garn
Winter 2016

As gold which he cannot spend
will make no man rich,
so knowledge which he cannot apply
will make no man wise.
Samuel Johnson: The Idler No. 84

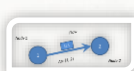
Learning Objectives

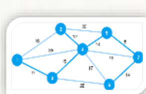
- To know some Graph Theory
  - Terminology network definition, cycles, …
- To understand the minimal spanning tree (MST)
  - To know the solution procedure
  - To state it as a Mathematical Program
  - To be aware of typical MST applications
- To create algorithms

Autumn 2016 ML&SCM 2

Contents

- Graph Theory
- Minimal Spanning Tree
- Applications

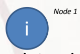




Autumn 2016 ML&SCM 3

Node

- Node = vertex

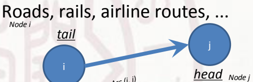


  - Cities, transshipment points, airports, warehouses, ...

Autumn 2016 ML&SCM Graph Theory

Arc
Terminology

- Arc = directed edge
  - Roads, rails, airline routes, ...

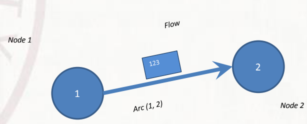


  - (*i,j*) *emanates* from *i* and
  - (*i,j*) *terminates* in *j*
  - (*i,j*) is *incident* to *i* and *j*
  - (*i,j*) is an *outgoing* arc of *i*
  - (*i,j*) is an *incoming* arc of j
  - If there is *(i,j)* the node *i* and *j* are *adjacent*

Autumn 2016 ML&SCM Graph Theory

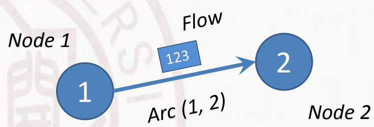
Flow

- Products, freight, intermodal containers, passengers, vehicles, materials, goods ...
- Quantity, weight, cost, distance, time, ...



Autumn 2016 ML&SCM Graph Theory

Graphs & networks

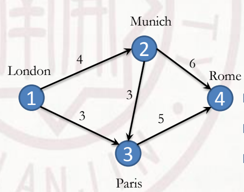


- *directed graph* $G = (N, A)$
  - consists of a set of nodes N and arcs A
- *directed network*
  - Is a directed graph and nodes and/or arcs have associated numerical values
- *subgraph* $G' = (N', A')$
  - if $N' \subseteq N$ and $A' \subseteq A$
- *spannning subgraph*
  - if $N' = N$ and $A' \subseteq A$

Autumn 2016 ML&SCM Graph Theory

Network example

- Four nodes N, five arcs A and five flows W.
- "London", node 1, termed ***origin or source***
- Arcs identified by beginning and ending node numbers.
- Flow assigned to each arc (distance, time, cost, …)



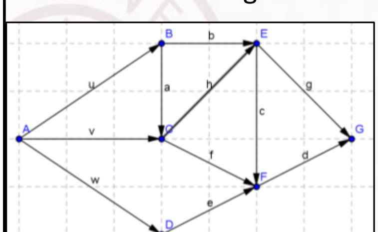
Network of Railroad routes

London, Munich, Paris, Rome

- $N = \{1, 2, 3, 4\}$
- $A = <(1, 2), (1, 3), (2, 3), (2, 4), (3, 4)>$
- $W = <4, 3, 3, 6, 5>$

ML&SCM Graph Theory

Node Degrees



What is the outdegree of A?
3
What is the outdegree of B?
2
What is the indegree of B?
1
What is the degree of B?
3

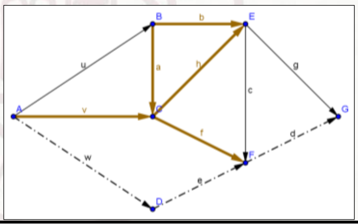
- *outdegree* = #outgoing arcs
- *indegree* = #incoming arcs
- *degree* = outdegree + indegree

Is the sum of all indgrees the same as the sum of all outdegrees?
yes

Autumn 2016 ML&SCM Graph Theory

Walks & paths

- A *walk* is a sequence of nodes and arcs n_1-a_1-n_2-a_2-…-n_{r-1}-a_{r-1}-n_r, such that nodes and arcs are a subgraph of (N,A)
- A *path* is a walk without any repetition of nodes.

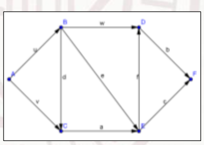


Is A-v-C-h-E-b-a-C-f-F a walk or path?
walk

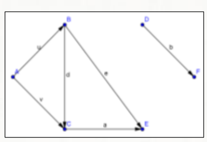
Is A-w-D-e-F-d-G a walk or path?
walk and path

Connected

- Two nodes are *connected* in a graph, if there is at least one path from node *i* to node *j*.
- A *graph* is *connected* if every pair of nodes is connected.



A and F are connected

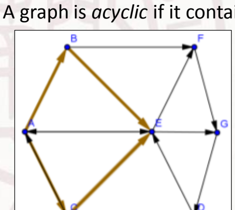


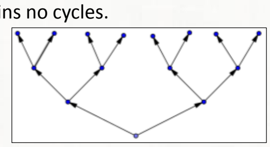
Graph not connected

Autumn 2016 ML&SCM

Cycles

- A *cycle* is a closed path of positive length. A cycle is a path $n_1 - n_2$ - … - n_r together with the arc (n_r, n_1) or (n_1, n_r).
- A graph is *acyclic* if it contains no cycles.





Binary tree is acyclic

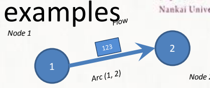
A-B-E-C-A is a cycle

Autumn 2016 ML&SCM

Forests & trees

- A *forest* is an acyclic graph. A graph that contains no cycles is a forest. A forest is a collection of trees.
- A *tree* is an acyclic connect graph. A tree is a connect graph that contains no cycles.

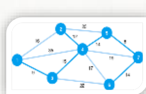
Graph Theory - examples



- Communication Networks
  - Nodes: Telephone exchanges, routers, satellites, computers, ...
  - Arcs: Cables, Fiber optic links, wireless connections
  - Flow: Data, Video, “voice”
- Computer & Electrical circuits
  - Nodes: Processors, gates, registers, intersections, ...
  - Arcs: Wires, resistors, capacitors, inductors, ...
  - Flow: electrical current
- Hydraulic Systems
  - Nodes: Lakes, reservoirs, pumping stations
  - Arcs: pipelines
  - Flow: water, gas, oil, ...

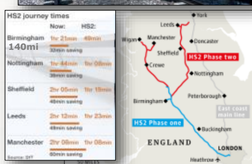
Contents

- Graph Theory
- Minimal Spanning Tree
- Applications



High Speed rail network

- Trains
  - 400m long; 1,100 seats;
  - 250mph (124mph~200km/h)
- Network
  - Distances: several 100 miles
  - Times: <1.5 hours
  - Cost: £42.6bn [June, 2013]
    - £14.4bn contingency

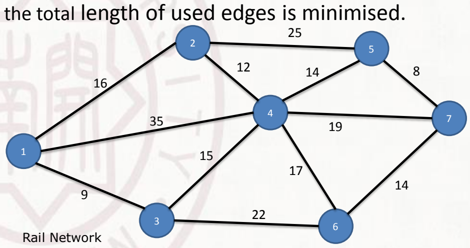


Sources: Guardian, BBC, Gov.uk

Autumn 2014 ML&SCM

MST Problem Specification

- All nodes have to be connected via edges such that the total length of used edges is minimised.



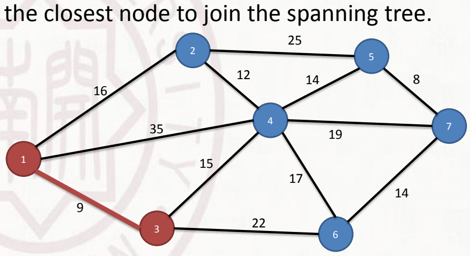
Rail Network

N = {1, 2, …, 7}
E = <(1,2),(1,3),…,(6,7)>
W = <16, 9, …, 14>

Autumn 2014 ML&SCM 17

MST – Finding a solution (1/6)

- Start with any node in the network and select the closest node to join the spanning tree.



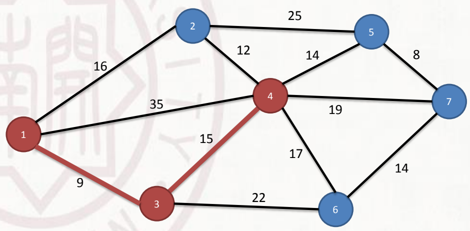
Spanning Tree with Nodes 1 and 3

Autumn 2014 ML&SCM 18

MST – Finding a solution (2/6)

Nankai University MBA中心

Select the closest node not presently in the spanning tree.



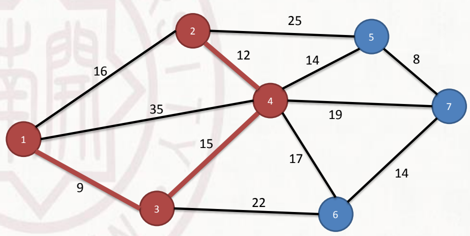
Spanning Tree with Nodes 1, 3, and 4

Autumn 2014 ML&SCM 19

MST – Finding a solution (3/6)

Nankai University MBA中心

Continue to select the closest node not presently in the spanning tree.



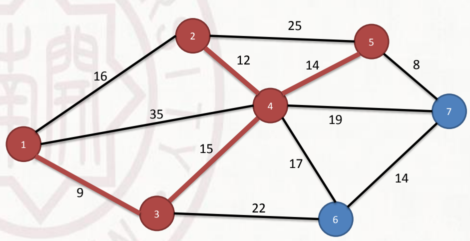
Spanning Tree with Nodes 1, 2, 3, and 4

Autumn 2014 ML&SCM 20

MST – Finding a solution (4/6)

Nankai University MBA中心

Continue to select the closest node not presently in the spanning tree.

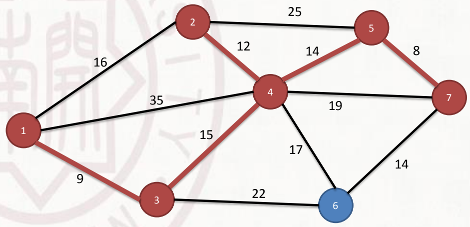


Spanning Tree with Nodes 1, 2, 3, 4, and 5

Autumn 2014 ML&SCM 21

MST – Finding a solution (5/6)

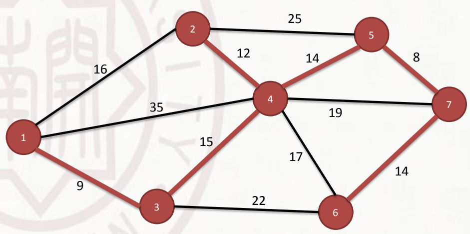
Continue to select the closest node not presently in the spanning area.



Spanning Tree with Nodes 1, 2, 3, 4, 5, and 7

MST – Finding a solution (6/6)

Optimal Solution



Minimal Spanning Tree for Rail network

MST – a solution procedure

1. Select any starting node (usually node 1)
2. Add the node closest to the starting node to form a tree
3. Select the closest node
   - from the remaining nodes that is not in the tree
   - and does not form a cycle
4. Repeat step 3 until all nodes are connected

Advantage: simple to formulate
Disadvantage: Step 3 is inefficient, because distances have to be repeatedly sorted

MST – Kruskal's approach (in words)

1. Sort all edges
2. Go through sorted edge list
   - i.e. take edge from top not used yet
3. Add the edge - if no cycle is formed
4. Stop when all nodes are connected, otherwise continue with 2

Advantage:
- Efficient algorithm,
- "Natural" clustering procedure

Autumn 2016 ML&SCM 25

Kruskal: Step 1 sort edges

Arcs

from	to	distance
1	2	16
1	3	9
1	4	35
2	4	12
2	5	25
3	4	15
3	6	22
4	5	14
4	6	17
4	7	19
5	7	8
6	7	14

1. Sort arcs by distances

from	to	distance
5	7	8
1	3	9
2	4	12
4	5	14
6	7	14
3	4	15
1	2	16
4	6	17
4	7	19
3	6	22
2	5	25
1	4	35

Quicksort is one of the fastest and most popular sorting algorithms

Autumn 2016 ML&SCM 26

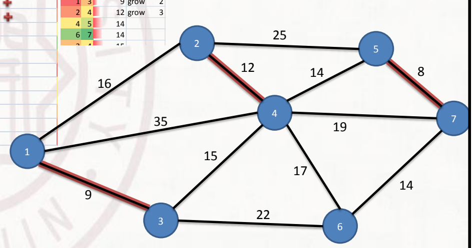
Kruskal – go through edge list

1. Sort arcs by distances

from	to	distance
5	7	8
1	3	9
2	4	12
4	5	14
6	7	14
3	4	15
1	2	16
4	6	17
4	7	19
3	6	22
2	5	25
1	4	35

2. Grow trees

from	to	distance	Step	Tree
5	7	8	grow	1
1	3	9	grow	2
2	4	12	grow	3
4	5	14		
6	7	14		



Shows how the algorithm forms naturally clusters

Autumn 2016 ML&SCM 27

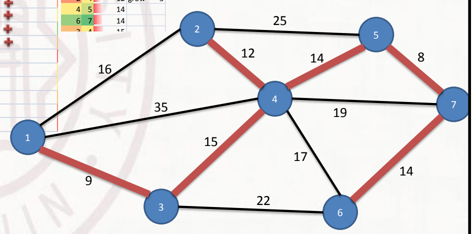
Kruskal – continue iterating through the edges

1. Sort arcs by distances

from	to	distance	
5	7	8	+
1	3	9	+
2	4	12	+
4	5	14	+
6	7	14	+
3	4	15	+
1	2	16	
4	6	17	
4	7	19	
3	6	22	
2	5	25	
1	4	35	

2. Grow trees

from	to	distance	Step	Tree
5	7	8	grow	1
1	3	9	grow	2
2	4	12	grow	3
4	5	14		
6	7	14		



Note that if edge (4,6) would have the distance 14.5, then adding the edge would form the cycle <4, 5, 7, 6, 4>, which we don't want!

Autumn 2016 ML&SCM 28

Kruskal's example summary

Arcs

from	to	distance
1	2	16
1	3	9
1	4	35
2	4	12
2	5	25
3	4	15
3	6	22
4	5	14
4	6	17
4	7	19
5	7	8
6	7	14

1. Sort arcs by distances

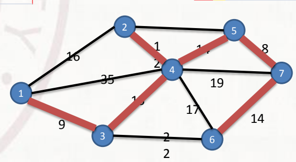
from	to	distance
5	7	8
1	3	9
2	4	12
4	5	14
6	7	14
3	4	15
1	2	16
4	6	17
4	7	19
3	6	22
2	5	25
1	4	35

2. Grow trees

from	to	distance	Step	Tree
5	7	8	grow	1
1	3	9	grow	2
2	4	12	grow	3
4	5	14		
6	7	14		

3. Continue growing trees

from	to	distance	Step	Tree	Nodes
5	7	8	grow tree	1	5,7
1	3	9	grow tree	2	1,3
2	4	12	grow tree	3	2,4
4	5	14	combine	1,3 >> 1	2,4,5,7
6	7	14	grow tree	1	2,4,5,6,7
3	4	15	combine	2,1 >> 1	1,2,3,4,5,6,7
1	2	16			
4	6	17			
4	7	19			
3	6	22			
2	5	25			
1	4	35			



Autumn 2016 ML&SCM 29

MST algorithm

- Input: Network (V, E, w)
  – connected graph with m nodes and n edges, $w_i \geq 0$

1. Edge ordering $e_1, e_2, \cdots, e_n$ such that $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_n)$. $E^0 = \emptyset, k = 1$
2. If $H = \left(V, E^{k-1} \cup \{e_k\}\right)$ is acyclic, then $E^k := E^{k-1} \cup \{e_k\}$, otherwise $E^k := E^{k-1}$
3. If $\left|E^k\right| = m - 1$, stop, $\left(V, E^k\right)$ is a spanning tree. Otherwise $k := k + 1$, and return to step 2

Source: Integer and combinatorial optimization - Nemhauser, George L., Wolsey, Laurence A. c1988 (I.3. p60ff)

Autumn 2016 ML&SCM 30

MST Algorithm (acyclic)

Kruskal's Algorithm (1956); Dijkstra introduced a solution in 1959

INPUT: *weighted arcs* A
OUTPUT: *minimal spanning tree* T
$A := \text{sort}_w A$ sort acrs by increasing weight
$\forall \quad t \in N : T^t = (T_N^t := t, T_A^t := \emptyset)$ each node is a tree
$i := 0$ arc index
$k := 0$ number of found arcs
while $k < |N| - 1$ not all arcs are found
    $i := i + 1$ increment arc index
    $I := T_N^{\{t\}} \cap_I A_i$ find tree(s) including arc nodes
    $|I| = 2 \Rightarrow$ if two trees then
        $T^{I_1} := T^{I_1} \cup T^{I_2} \cup A_i,$ join the trees
        $T^{I_2} := \emptyset$ destroy not needed tree
        $k := k + 1$ increment number of found arcs

31

MST and LP

- x_{ij} … 0-1 variable whether arc (i,j) is selected as part of the MST
- Minimize $\sum_{(i,j)\in A} c_{ij} x_{ij}$
- subject to
  - $\sum_{(i,j)\in A} x_{ij} = n - 1,$
    - cardinality constraint, exactly n-1 arcs
  - $\sum_{(i,j)\in A(S)} x_{ij} = |S| - 1$ for any S of nodes
    - "packing" constraint, ensure no cycles

exponential number of constraints

Autumn 2016 ML&SCM 32

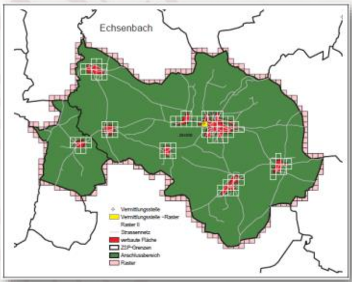
Contents

- Graph Theory
- Minimal Spanning Tree
- Applications



Autumn 2016 ML&SCM 33

Application – Access Network

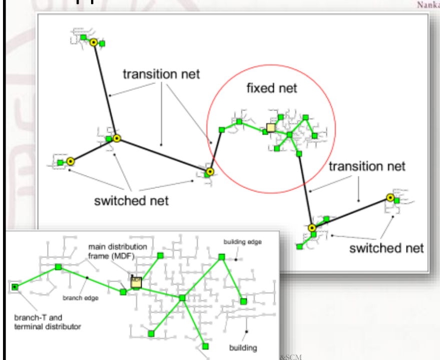


Optimal trenching and cabling
- Trenching 80% of costs

Green field approach

Autumn 2016 ML&SCM 34

Application – Access Network



35

Rail network

- What is the minimal required configuration?

Autumn 2016 ML&SCM 36

Recent applications

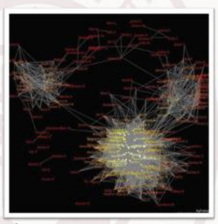
南開大學 Nankai University | MBA中心

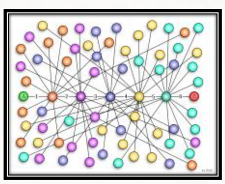

Mirror NEWS
UK weather: 220,000 homes without power across England as Storm St Jude lashes country

- What's the most effective way to re-establish the electrical network?

More networks

南開大學 Nankai University | MBA中心


Social networks


Six degrees of separation
Barabási, Albert-László. 2003.
Linked: How Everything is Connected to Everything Else and What It Means for Business, Science, and Everyday Life.

Recap – "the end is near"

南開大學 Nankai University | MBA中心

- What is a graph?
  - Nodes and Arcs
  - TV network
- What is a network?
  - A graph and arcs have values ("weights").
  - National grids, telecommunication, social networks
- What is a minimal spanning tree?
  - "The least number of arcs that is necessary to reach all nodes and minimises the total weight."

The End

- Any questions?

References/Literature

- Introduction to management science: a modeling and case studies approach with spreadsheets - Hillier, Frederick S., Hillier, Mark S., Schmedders, Karl, Stephens, Molly
  - Chapter: 6 (p194ff)
- Introduction to operations research - Hillier, Frederick S., Lieberman, Gerald J. 2010
  - Chapter: 9 (p358ff)
- Network flows: theory, algorithms, and applications - Ahuja, Ravindra K., Magnanti, Thomas L., Orlin, James B. c1993
  - Chapter: 13 (p510ff)
- Issues in operations management: [a management science approach] - Taylor, Bernard W., Garn, Wolfgang 2011 (p92ff)
- Integer and combinatorial optimization - Nemhauser, George L., Wolsey, Laurence A. c1988 (I.3. p60ff)

Defintions

- *Nodes* are natural numbers.
- An *arc* is an ordered pair of nodes.
- An *edge* is an unordered pair of nodes.
- A *direct graph* consists of a set of nodes and a set of arcs whose elements are ordered pairs of distinct nodes.
- An *undirect graph* consists of a set of nodes and a set of arcs whose elements are unordered pairs of distinct nodes.
- A *walk* is a sequence of nodes and arcs n_1-a_1-n_2-a_2-…-n_{r-1}-a_{r-1}-n_r, such that nodes and arcs are a subgraph of (N,A)
- A *path* is a walk without any repetition of nodes.
- Two nodes are connected in a graph, if there is at least one path from node i to node j.
- A *graph* is connected if every pair of nodes is connected.
- A *cycle* is a closed path of positive length. A cycle is a path $n_1 – n_2$ - … - n_r together with the arc (n_r, n_1) or (n_1, n_r).
- A graph is *acyclic* if it contains no cycles.
- A *forest* is an acyclic graph. A graph that contains no cycles is a forest. A forest is a collection of trees.
- A *tree* is an acyclic connect graph. A tree is a connect graph that contains no cycles.