Multi net routing: Steiner Min/Max Trees

Sowmiya Sivanandham

Vidyasagar Mukala

Georgia Institute of Technology

Presentation Outline

- Overview of SMMT algorithm
- SMMT Phase – Prim’s algorithm
- SP Phase – Floyd’s algorithm
- Implementation details
- Experimental Results
- Demo

Overview of SMMT

- Routes multiple nets one-by-one
- Rip-up and re-routing
- Order in which nets are routed is important
- Given an undirected, edge-weighted graph G(V,E) and a net n that contains a subset of nodes D, SMMT of n is a Steiner tree of n, where maximum weight among all edges in the tree is minimized
- Edge weight is equivalent to usage which reflects congestion

SMMT Cont'd

- **SMMT phase (Reduces the maximum weight among all edges)**

  For each un-routed net from list of ordered nets,

  - **Build MST for the routing graph**
  - **Remove 1-degree Steiner nodes which results in SMMT**
  - **If wirelength of SMMT is less than (cj * HPBB), update the routing graph with the SMMT, else discard it.**

  Multiple passes helps routing nets that failed in the previous passes

- **SP Phase (Reduces wire length)**

  For each net from list of ordered nets,

  - **Rip-up from routing graph**
  - **Build Shortest Path tree between nodes of the net**
  - **If wirelength of SP tree is less than wirelength of SMMT, accept SP tree, else discard it.**

MST – Prim's Algorithm

Consider weighted graph G = (V, E)

Begin with U = {1}

Add one edge from (V-U) to U at a time

Find the shortest edge that connects U and (V-U) and add the vertex to U

Repeat until U = V

Implementation – Prim's

Graph is represented using Adjacency Matrix

C[u][v] - Cost of going from node 'u' to node 'v'

Lowcost[x] - Contains lowest cost through which nodes in U-V are connected to U

Closest[x] – Contains the node in {U} that is closest to node'x'

Initialize Lowcost[i] = C[1,i]

Closest[i] = 1 // U = {1}

min = Min (Lowcost (2 to n)) //say kth node has Low cost

Add node k to U and make Lowcost[k] = ∞

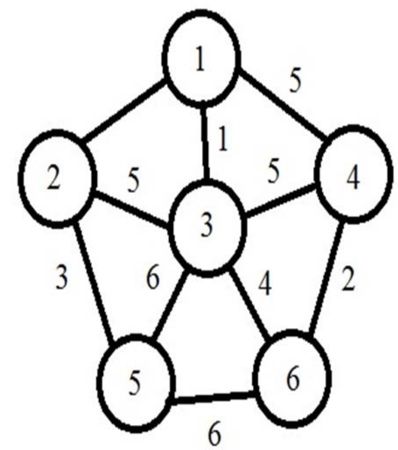
Update Lowcost and Closest Arrays

```
for (j = 2 to n)
        If ( C[k,j] < Lowcost[j] )
        Lowcost[j] = C[k,j]
        Closest[j] = k
end
```

Georgia Institute of Technology

Prim's Algorithm - Example

C[u][v]	1	2	3	4	5	6
1	∞	6	1	5	∞	∞
2	6	∞	5	∞	3	∞
3	1	5	∞	5	6	4
4	5	∞	5	∞	∞	2
5	∞	3	6	∞	∞	6
6	∞	∞	4	2	∞	6



Graph (V,E)

Example Cont'd

U = {1}	2	3	4	5	6
Lowcost	6	1	5	∞	∞
Closest	1	1	1	1	1

min = 1; k = 3

U = {1,3}	2	3	4	5	6
Lowcost	**5**	∞	5	**6**	**4**
Closest	**3**	1	1	**3**	**3**

min = 4; k = 6

U = {1,3,6}	2	3	4	5	6
Lowcost	5	∞	**2**	6	∞
Closest	3	1	**6**	3	3

min = 2; k = 4

Georgia Institute of Technology

Example Cont'd

U = {1,3,6,4}	2	3	4	5	6
Lowcost	5	∞	∞	6	∞
Closest	3	1	6	3	3

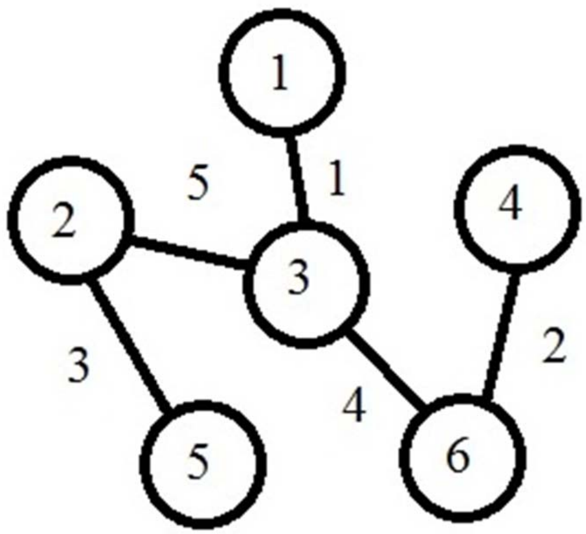
min = 5 ; k = 2

U = {1,3,6,4,2}	2	3	4	5	6
Lowcost	∞	∞	∞	**3**	∞
Closest	3	1	6	**2**	3

min = 3 ; k = 5

Example Cont'd

U = {1,3,6,4,2,5}	2	3	4	5	6
Lowcost	∞	∞	∞	∞	∞
Closest	3	1	6	**2**	3



MST of Graph (V,E)

SP – Floyd's Algorithm $O(n^3)$

All pairs Shortest Path Solution

C[u][v] - Represents the cost of going from node 'u' to node 'v'

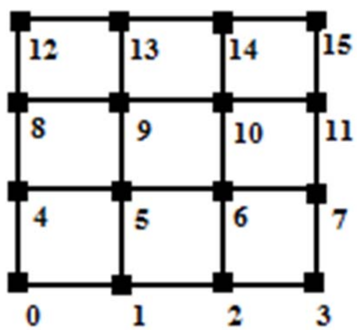
A[u][v] – Shortest distance to go from node 'u' to 'v'

P[u][v] – Stores information of intermediate node connecting node 'u' and node 'v'

```
for (i = 1 to n)
    for(j = 1 to n)
          A[i][j] = C[i][j]
          P[i][j] = 0
     end
end
for(i = 1 to n)
   A[i][i] = 0
end
for (k = 1 to n)
  for (i = 1 to n)
    for(j = 1 to n)
       if( A[i][k] + A[k][j] < A[i][j]) then begin
       A[i][j] = A[i][k] +A[k][j]
       P[i][j] = k
    end
  end
end
```

Implementation Details

Routing Grid



Adjacency Matrix for Routing Grid

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	∞	0	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	∞	0	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	∞	0	∞	0	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	∞	∞	0	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	∞
4	0	∞	∞	∞	∞	0	∞	∞	0	∞	∞	∞	∞	∞	∞	∞
5	∞	0	∞	∞	0	∞	0	∞	∞	0	∞	∞	∞	∞	∞	∞
6	∞	∞	0	∞	∞	0	∞	0	∞	∞	0	∞	∞	∞	∞	∞
7	∞	∞	∞	0	∞	∞	0	∞	∞	∞	∞	0	∞	∞	∞	∞
8	∞	∞	∞	∞	0	∞	∞	∞	∞	0	∞	∞	0	∞	∞	∞
9	∞	∞	∞	∞	∞	0	∞	∞	0	∞	0	∞	∞	0	∞	∞
10	∞	∞	∞	∞	∞	∞	0	∞	∞	0	∞	0	∞	∞	0	∞
11	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	0	∞	∞	∞	∞	0
12	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	0	∞	∞
13	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	0	∞	0	∞
14	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	0	∞	0
15	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	0	∞

Georgia Institute of Technology

Implementation Cont'd

- Use of 3 states for weights:
  - INF → Edge cannot exist
  - 0 → Edge can exist but no net has been routed
  - n → Edge can exist and has nets routed along that edge
- Accessing each node in the routing grid
  - k = x*MAX + y

  Where,
    - (x,y) -> co-ordinate of the vertex
    - MAX -> Maximum size of routing grid
- Size of routing grid
  - Use of MAX and actual_MAX (Static arrays)

Experimental Analysis

Number of nets	Number of terminals	Grid size	Wire length	Maximum Edge usage
10	31	8	71	3
20	67	10	258	5
30	87	13	313	7
50	133	16	465	18

Inference

- **Routing results are dependent on the constraint cj**
  - Smaller cj → tighter constraint → Routing efficiency decreases
- **Congestion is dependent on grid size**
  - Smaller grid with more nets increases congestion(increases maximum edge weight)
  - SMMT tries to reduce the usage (reduces congestion)
  - SP tries to reduce wire length at the cost of usage (increases congestion)
  - Multiple passes increases routing efficiency at the cost of runtime
    - During each pass, the underlying routing grid is different, although the same net is routed again
    - Hence, those nets for which routing failed in the previous passes also become routable

Routing Efficiency Vs Passes (cj = 1)

- **SMMT Pass 1:**
- Routing in SMMT phase for net 0: **not accepted**
- Routing in SMMT phase for net 1: **not accepted**
- Routing in SMMT phase for net 2: **not accepted**
- Routing in SMMT phase for net 3: **not accepted**
- Routing in SMMT phase for net 4: accepted
- Routing in SMMT phase for net 5: **not accepted**
- Routing in SMMT phase for net 6: **not accepted**
- Routing in SMMT phase for net 7: accepted
- Routing in SMMT phase for net 8: **not accepted**
- Routing in SMMT phase for net 9: **not accepted**
- **SMMT Pass 2:**
- Routing in SMMT phase for net 0: **not accepted**
- Routing in SMMT phase for net 1: **not accepted**
- Routing in SMMT phase for net 2: **not accepted**
- Routing in SMMT phase for net 3: **not accepted**
- Routing in SMMT phase for net 4: accepted
- Routing in SMMT phase for net 5: **not accepted**
- Routing in SMMT phase for net 6: accepted
- Routing in SMMT phase for net 7: accepted
- Routing in SMMT phase for net 8: accepted
- Routing in SMMT phase for net 9: **not accepted**

Georgia Institute of Technology

Illustration Cont'd

SMMT Pass 3:
Routing in SMMT phase for net 0: **not accepted**
Routing in SMMT phase for net 1: **not accepted**
Routing in SMMT phase for net 2: **not accepted**
Routing in SMMT phase for net 3: **not accepted**
Routing in SMMT phase for net 4: accepted
Routing in SMMT phase for net 5: **not accepted**
Routing in SMMT phase for net 6: accepted
Routing in SMMT phase for net 7: accepted
Routing in SMMT phase for net 8: accepted
Routing in SMMT phase for net 9: **not accepted**
SP Phase:
Routing in SMMT phase for net 0: **not accepted**
Routing in SMMT phase for net 1: accepted
Routing in SMMT phase for net 2: **not accepted**
Routing in SMMT phase for net 3: accepted
Routing in SMMT phase for net 4: accepted
Routing in SMMT phase for net 5: accepted
Routing in SMMT phase for net 6: accepted
Routing in SMMT phase for net 7: accepted
Routing in SMMT phase for net 8: accepted
Routing in SMMT phase for net 9: accepted

Routing Efficiency Vs Passes (cj = 3)

SMMT Pass 1:
Routing in SMMT phase for net 0: accepted
Routing in SMMT phase for net 1: **not accepted**
Routing in SMMT phase for net 2: accepted
Routing in SMMT phase for net 3: accepted
Routing in SMMT phase for net 4: accepted
Routing in SMMT phase for net 5: accepted
Routing in SMMT phase for net 6: accepted
Routing in SMMT phase for net 7: accepted
Routing in SMMT phase for net 8: accepted
Routing in SMMT phase for net 9: accepted
SMMT Pass 2:
Routing in SMMT phase for net 0: accepted
Routing in SMMT phase for net 1: **not accepted**
Routing in SMMT phase for net 2: accepted
Routing in SMMT phase for net 3: accepted
Routing in SMMT phase for net 4: accepted
Routing in SMMT phase for net 5: accepted
Routing in SMMT phase for net 6: accepted
Routing in SMMT phase for net 7: accepted
Routing in SMMT phase for net 8: accepted
Routing in SMMT phase for net 9: accepted

Illustration Cont'd

SMMT Pass 3:

Routing in SMMT phase for net 0: accepted
Routing in SMMT phase for net 1: **not accepted**
Routing in SMMT phase for net 2: accepted
Routing in SMMT phase for net 3: accepted
Routing in SMMT phase for net 4: accepted
Routing in SMMT phase for net 5: accepted
Routing in SMMT phase for net 6: accepted
Routing in SMMT phase for net 7: accepted
Routing in SMMT phase for net 8: accepted
Routing in SMMT phase for net 9: accepted

SP Phase:

Routing in SMMT phase for net 0: accepted
Routing in SMMT phase for net 1: accepted
Routing in SMMT phase for net 2: accepted
Routing in SMMT phase for net 3: accepted
Routing in SMMT phase for net 4: accepted
Routing in SMMT phase for net 5: accepted
Routing in SMMT phase for net 6: accepted
Routing in SMMT phase for net 7: accepted
Routing in SMMT phase for net 8: accepted
Routing in SMMT phase for net 9: accepted

Georgia Institute of Technology

SMMT Vs SP

	SMMT PHASE		SP PHASE	
NET	MAX E WGT	WIRELENGTH	MAX E WGT	WIRELENGTH
0	1	N/R	1	N/R
1	1	N/R	2	17
2	1	N/R	1	N/R
3	1	N/R	2	25
4	1	4	1	4
5	1	N/R	2	11
6	1	25	2	7
7	1	5	2	5
8	1	9	1	7
9	1	N/R	1	6

Georgia Institute of Technology

Results –Phase I (SMMT)

```
SMMT - Phase I Net #1
(1,1) (2,2) (4,0) (4,4)

(20 )      (21 )      (22 )      (23 )      (24 )
                                               |
                                               |
                                               1
                                               |
                                               |
(15 )      (16 )      (17 )-- 1--(18 )-- 1--(19 )
                         |
                         |
                         1
                         |
                         |
(10 )      (11 )      (12 )      (13 )      (14 )
                         |
                         |
                         1
                         |
                         |
(5  )-- 1--(6  )      (7  )      (8  )      (9  )
   |                     |
   |                     |
   1                     1
   |                     |
   |                     |
(0  )-- 1--(1  )-- 1--(2  )-- 1--(3  )-- 1--(4  )
```

Net 1 result after Phase I

Results – Phase II (SP)

```
Net 1:WL of SP(9) is less than WL of MST(12)

SHORTEST PATH - PHASE II
(20 )      (21 )      (22 )      (23 )      (24 )
                                               |
                                               |
                                               1
                                               |
                                               |
(15 )      (16 )      (17 )      (18 )      (19 )
                                               |
                                               |
                                               1
                                               |
                                               |
(10 )      (11 )      (12 )-- 1--(13 )-- 1--(14 )
                         |
                         |
                         1
                         |
                         |
(5  )      (6  )-- 1--(7  )      (8  )      (9  )
                         |
                         |
                         1
                         |
                         |
(0  )      (1  )      (2  )-- 1--(3  )-- 1--(4  )
```

Net 1 result after Phase II

Results – Final Grid

```
GRID

(20 )-- 0--(21 )-- 0--(22 )-- 0--(23 )-- 0--(24 )
   |          |          |          |          |
   |          |          |          |          |
   1          0          1          1          1
   |          |          |          |          |
   |          |          |          |          |
(15 )-- 0--(16 )-- 1--(17 )-- 0--(18 )-- 0--(19 )
   |          |          |          |          |
   |          |          |          |          |
   2          1          0          2          2
   |          |          |          |          |
   |          |          |          |          |
(10 )-- 2--(11 )-- 1--(12 )-- 2--(13 )-- 2--(14 )
   |          |          |          |          |
   |          |          |          |          |
   1          2          2          1          1
   |          |          |          |          |
   |          |          |          |          |
(5  )-- 0--(6  )-- 3--(7  )-- 2--(8  )-- 1--(9  )
   |          |          |          |          |
   |          |          |          |          |
   1          2          2          0          1
   |          |          |          |          |
   |          |          |          |          |
(0  )-- 2--(1  )-- 1--(2  )-- 2--(3  )-- 2--(4  )
```

Final Grid

Thank You !