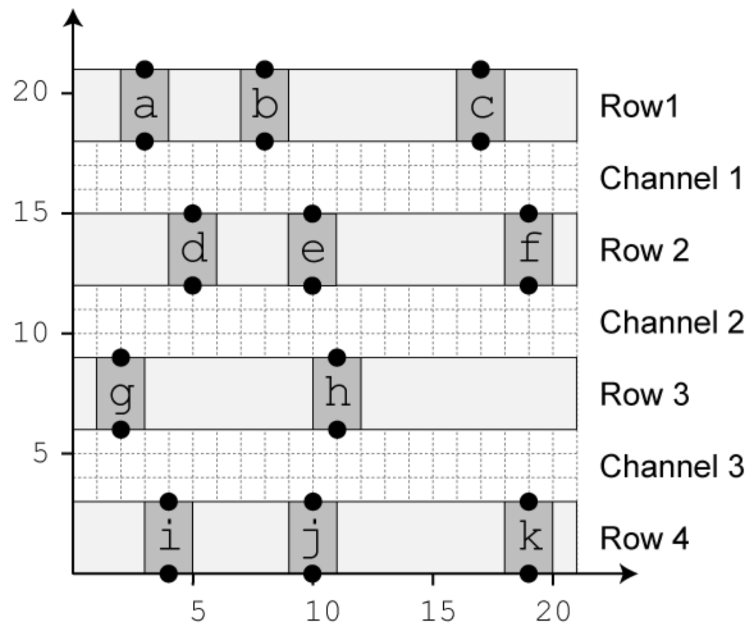
Iterative Deletion Routing Algorithm

- Perform routing based on the following placement
  - Two nets: $n_1 = \{b,c,g,h,i,k\}$, $n_2 = \{a,d,e,f,j\}$
  - Cell/feed-through width = 2, height = 3
  - Shift cells to the right, each cell contains self-feed-through



Feed-through Insertion

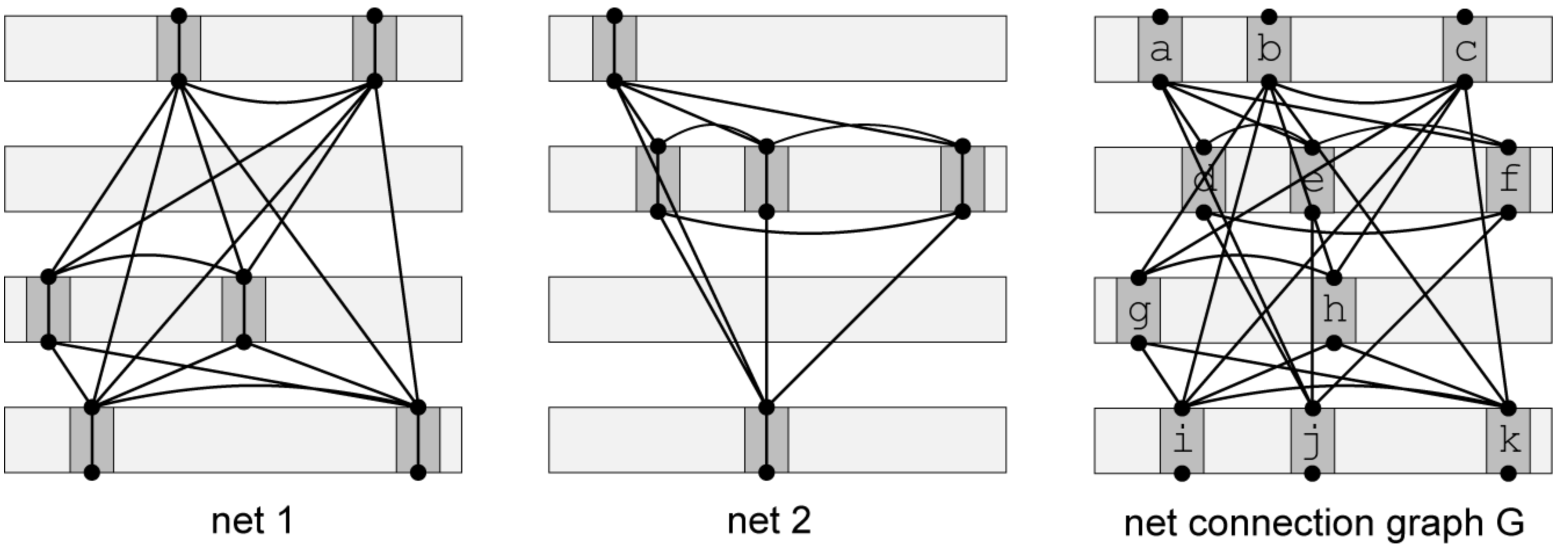
- Add one edge with min-weight at a time
  - Continue until we form a spanning forest
  - Our spanning forest needs 4+5 edges (why?)
  - Use K = 0.5

$$w(e(i,j)) = |x_i - x_j| + K \cdot \sum_{e \cap R_k \neq \emptyset} width(R_k)$$

  - Break ties in alphabetical order
  - Place feed-throughs right below top gate

Feed-through Insertion (cont)

- First step: build net connection graph
  - Union of individual complete graphs

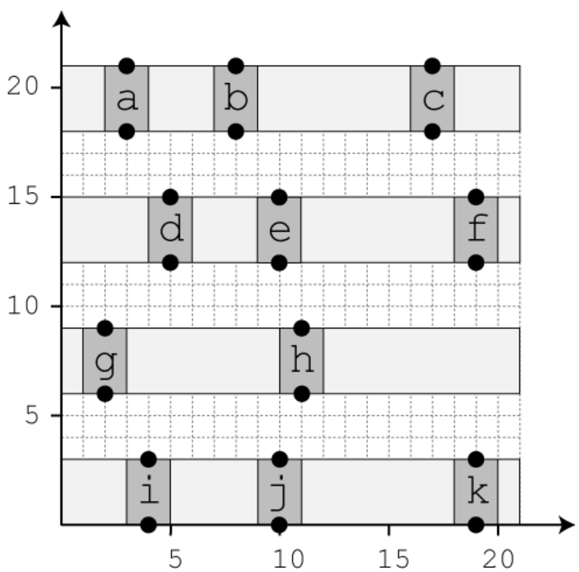


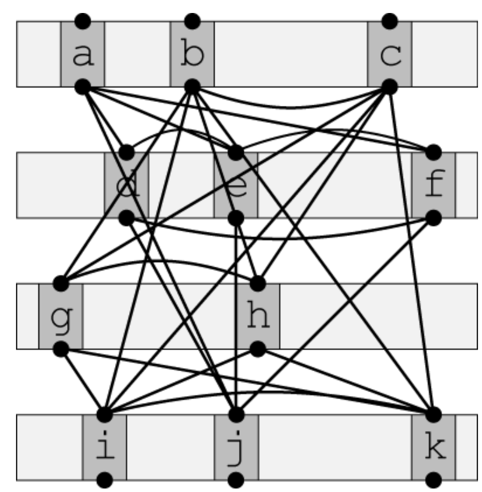
Feed-through Insertion (cont)

- Edge weight computation

$$w(e(i,j)) = |x_i - x_j| + K \cdot \sum_{e \cap R_k \neq \emptyset} width(R_k)$$

  - $w(a,d) = 2 + 0.5 \cdot 0 = 2$
  - $w(c,i) = 13 + 0.5 \cdot (21 + 21) = 34$





Feed-through Insertion (cont)

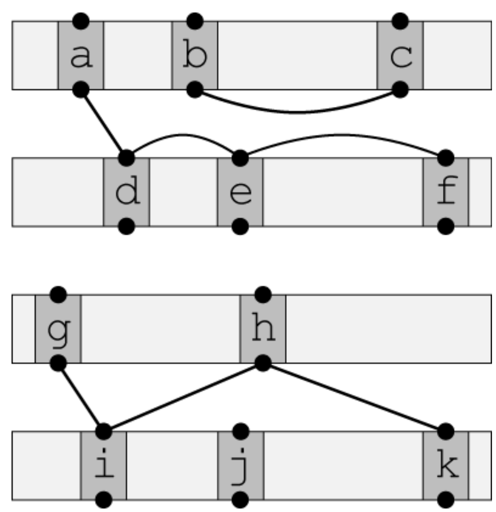
- Sorted edge list (increasing order)

edge	$\|x_i - x_j\|$	R_i	$w(e)$
(a, d)	2	-	$2 + 0.5(0) = 2$
(g, i)	2	-	$2 + 0.5(0) = 2$
(d, e)	5	-	$5 + 0.5(0) = 5$
(a, e)	7	-	$7 + 0.5(0) = 7$
(h, i)	7	-	$7 + 0.5(0) = 7$
(h, k)	8	-	$8 + 0.5(0) = 8$
....			
....			
(c, k)	2	R_2, R_3	$2 + 0.5(21 + 21) = 23$
(b, i)	4	R_2, R_3	$4 + 0.5(21 + 21) = 25$
(c, g)	15	R_2	$15 + 0.5(21) = 25.5$
(a, j)	7	R_2, R_3	$7 + 0.5(21 + 21) = 28$
(b, k)	11	R_2, R_3	$11 + 0.5(21 + 21) = 32$
(c, i)	13	R_2, R_3	$13 + 0.5(21 + 21) = 34$

Iterative Addition

- Adding first 7 edges
  - Based on increasing order of edge weight (should not form cycle)
  - Edge weight changes if feed-through is added
    - No feed-through is used for the first 7 edges, so no update

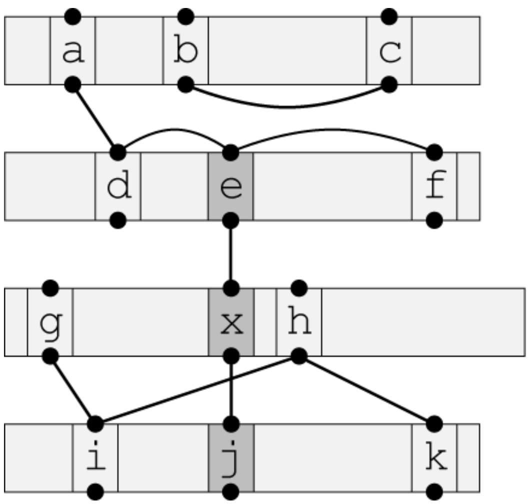
edge	$\lvert x_i - x_j \rvert$	R_i	$w(e)$	action
(a,d)	2	-	$2+0.5(0)=2$	added
(g,i)	2	-	$2+0.5(0)=2$	added
(d,e)	5	-	$5+0.5(0)=5$	added
(a,e)	7	-	$7+0.5(0)=7$	cycle
(h,i)	7	-	$7+0.5(0)=7$	added
(h,k)	8	-	$8+0.5(0)=8$	added
(b,c)	9	-	$9+0.5(0)=9$	added
(e,f)	9	-	$9+0.5(0)=9$	added
(g,h)	9	-	$9+0.5(0)=9$	cycle



Iterative Addition (cont)

- Adding 8th edge
  - Choose (e,j): does not create a cycle
  - Need a feed-through (= x) in third row (= R_3)
    - Some edges will have new weights (details in next slide)

edge	$\|x_i - x_j\|$	R_i	$w(e)$
(e, j)	0	R_3	$0 + 0.5(21) = 10.5$
(b, h)	3	R_2	$3 + 0.5(21) = 13.5$
(d, f)	14	-	$14 + 0.5(0) = 14$
....			
....			
(a, j)	7	R_2, R_3	$7 + 0.5(21 + 21) = 28$
(b, k)	11	R_2, R_3	$11 + 0.5(21 + 21) = 32$
(c, i)	13	R_2, R_3	$13 + 0.5(21 + 21) = 34$



Iterative Addition (cont)

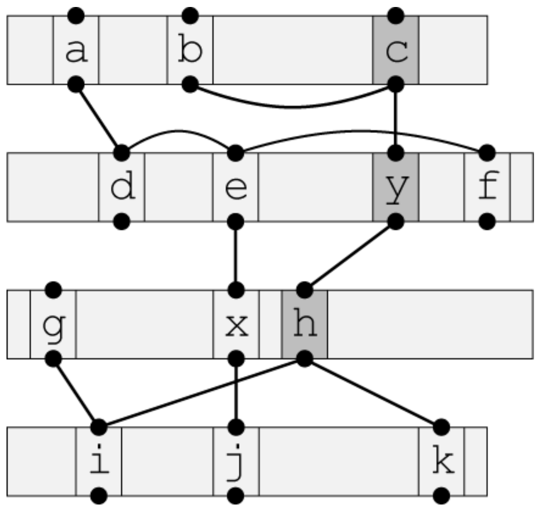
- Edge weight update after adding 8th edge
  - All edges intersecting with R_3
  - All edges connecting to cell h (because h is shifted)

	before	**after**			
edge	$w(e)$	$\lvert x_i - x_j \rvert$	R_i	$w(e)$	action
(b,h)	13.5	5	R_2	$5 + 0.5(21) = 15.5$	updated
(d,f)	14	14	-	$14 + 0.5(0) = 14$	
(i,k)	15	15	-	$15 + 0.5(0) = 15$	
(d,j)	15.5	5	R_3	$5 + 0.5(23) = 16.5$	updated
....					
....					
(c,g)	25.5	15	R_2	$15 + 0.5(21) = 25.5$	
(a,j)	28	7	R_2, R_3	$7 + 0.5(21 + 23) = 29$	updated
(b,k)	32	11	R_2, R_3	$11 + 0.5(21 + 23) = 33$	updated
(c,i)	34	13	R_2, R_3	$13 + 0.5(21 + 23) = 35$	updated

Iterative Addition (cont)

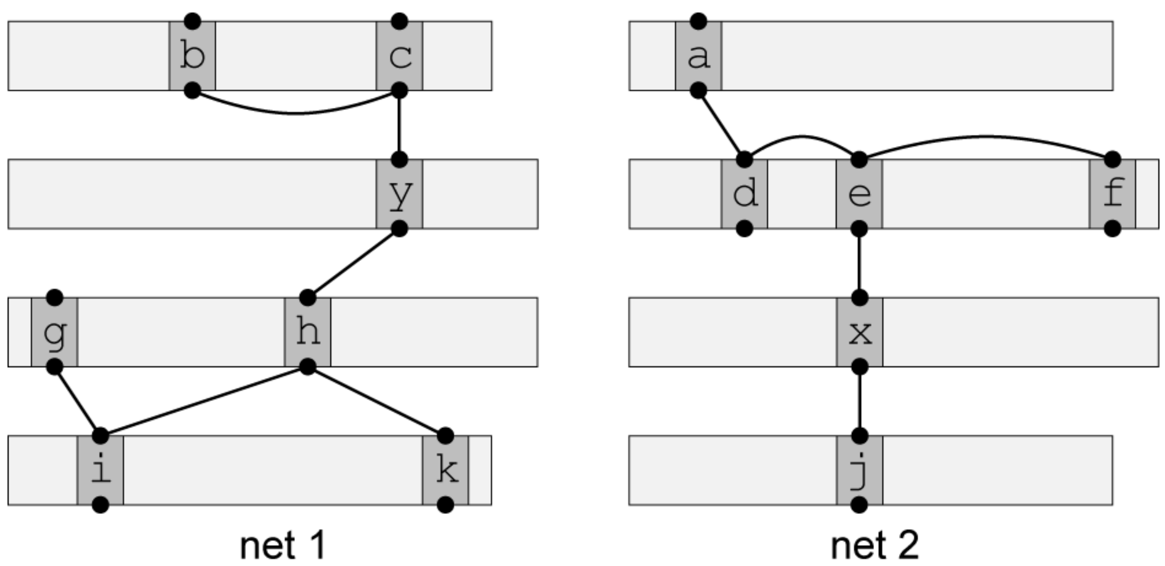
- Adding 9th (= last) edge
  - Skip (d,f) (= creates a cycle), so add (c,h)
  - Need a feed-through (= y) in R_2

edge	$w(e)$	action
(d, f)	14	cycle
(c, h)	14.5	added
(i, k)	15	
(b, h)	15.5	
....		
....		
(b, i)	26	
(a, j)	29	
(b, k)	33	
(c, i)	35	



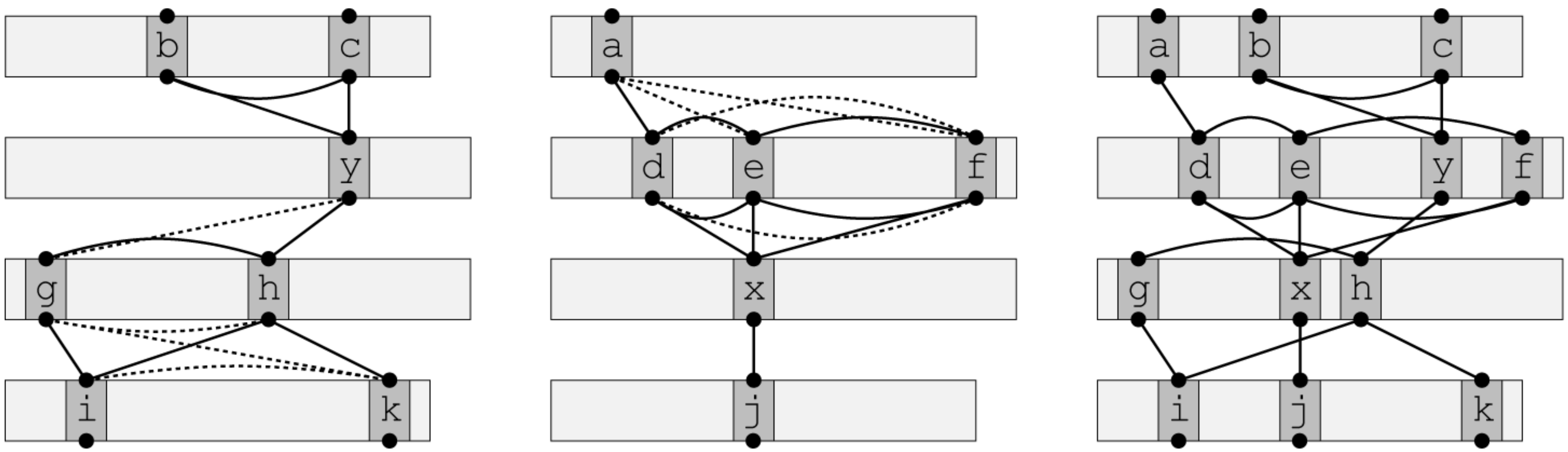
Iterative Addition (cont)

- Final Result
  - Two feed-throughs are inserted: already have routing solutions
  - Why do we need iterative deletion then?
    - Improve congestion

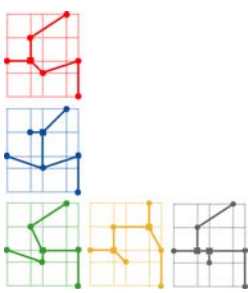


Iterative Deletion

- Step 1: obtain simplified net connection graph
  - Form cliques among pins in the same channel
  - Remove edges that connect non-adjacent pins (= dotted lines)

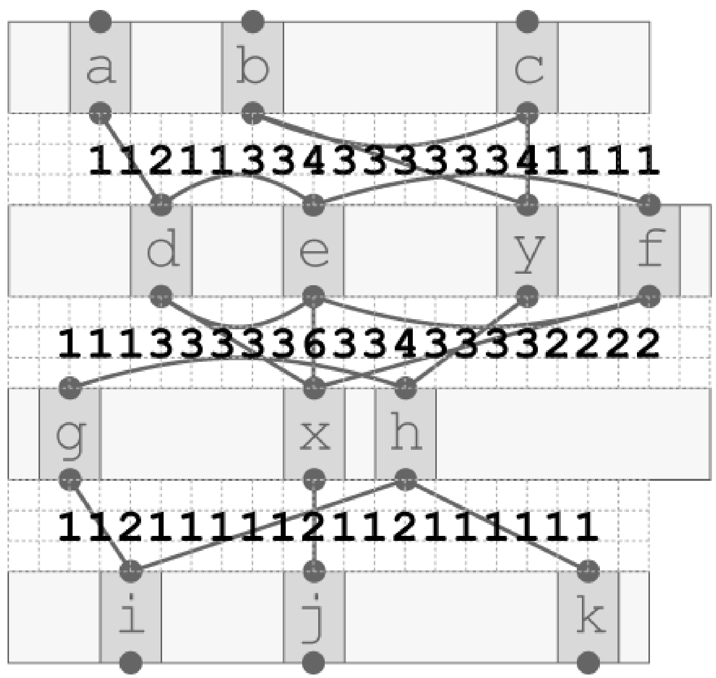


net 1 net 2 simplified net connection graph G'



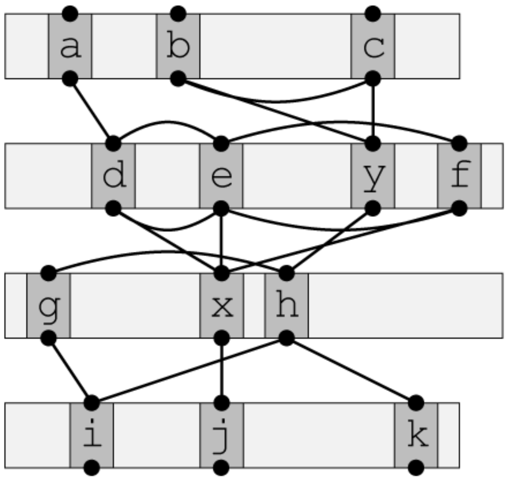
Iterative Deletion (cont)

- Step 2: compute channel density (= congestion)
  - Number of edges passing, beginning, or ending at each column
  - Density of channel 1/2/3 is 4/6/2 (= max value)



Iterative Deletion (cont)

- Step 3: delete edges in G'
  - Continue until we obtain spanning forest of G'
    - Should not isolate any node
  - Delete edges with max-weight first
    - $w(e) = d(e) / d(C_e)$
  - Break ties: delete edges
    - With longer x-span first
    - With higher edge density, $d(e)$
    - From bottom-most channel
    - Lexicographically

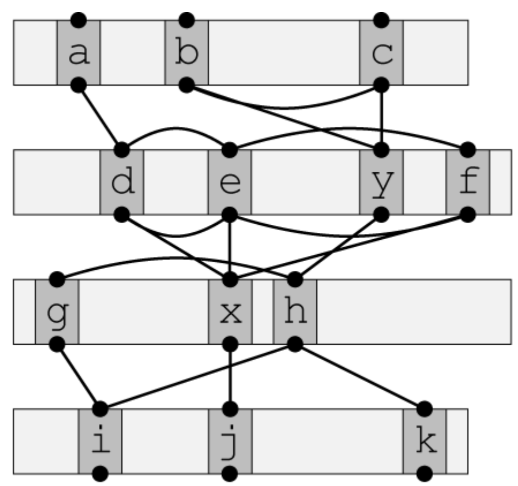


simplified net connection graph G'

Iterative Deletion (cont)

- Deleting first edge
  - Choose (*x*,*f*): does not isolate any node
  - Density of channel 2 reduces to 5:
    - weights of all edges in channel 2 to change

edge	x-span	$d(e)$	C_i	$w(e)$
(x, f)	11	6	C_2	$6/6 = 1$
(g, h)	11	6	C_2	$6/6 = 1$
(e, f)	11	6	C_2	$6/6 = 1$
(e, f)	11	4	C_1	$4/4 = 1$
....				
....				
(c, y)	0	4	C_1	$4/4 = 1$
(x, j)	0	2	C_3	$2/2 = 1$
(h, y)	4	4	C_2	$4/6 = 0.67$
(a, d)	2	2	C_1	$2/4 = 0.5$

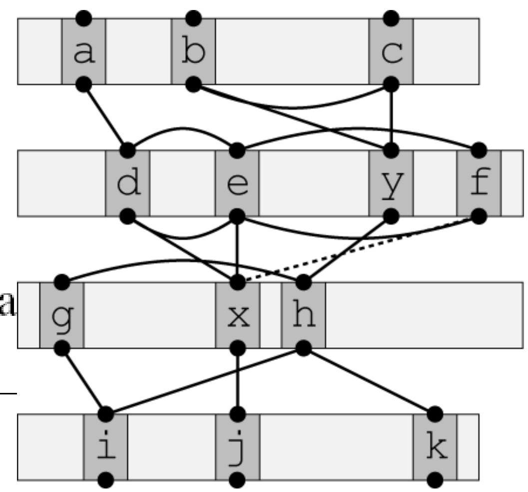


simplified net connection graph G'

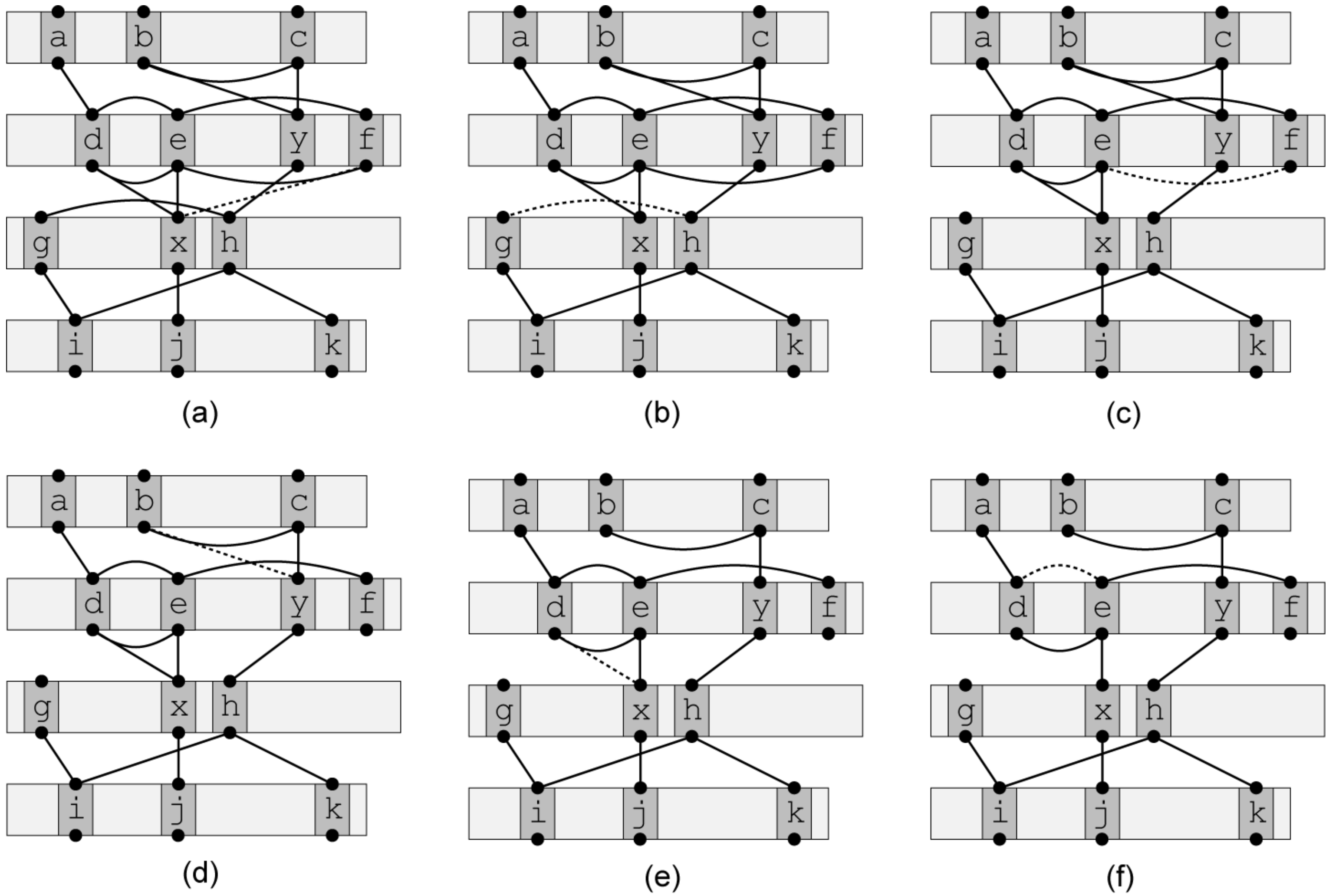
Iterative Deletion (cont)

- Edge weight update after deleting first edge
  - all edges in channel 2 to change

	before			after		
edge	$w(e)$	x-span	$d(e)$	C_i	$w(e)$	action
(g, h)	1	11	5	C_2	$5/5 = 1$	updated
(e, f)	1	11	5	C_2	$5/5 = 1$	updated
(e, f)	1	11	4	C_1	$4/4 = 1$	
(b, y)	1	9	4	C_1	$4/4 = 1$	
....						
....						
(c, y)	1	0	4	C_1	$4/4 = 1$	
(x, j)	1	0	2	C_3	$2/2 = 1$	
(h, y)	0.67	4	3	C_2	$3/5 = 0.6$	upda
(a, d)	0.5	2	2	C_1	$2/4 = 0.5$	



Iterative Deletion (cont)

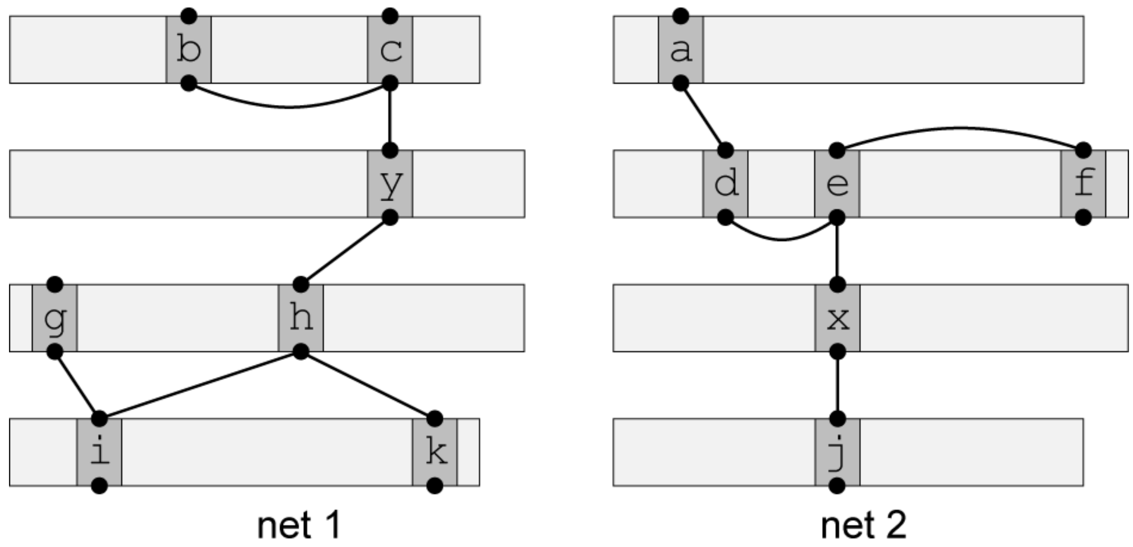


(a) (b) (c)

(d) (e) (f)

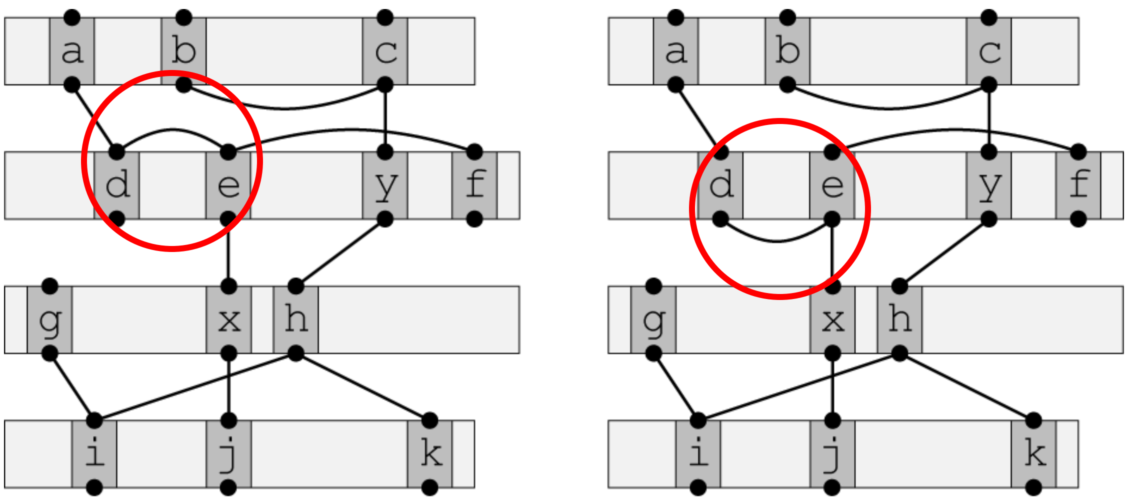
Iterative Deletion (cont)

- Final result



Iterative Addition vs Deletion

- Density of channel (= congestion) improved
  - Reduced from 3 to 2 in channel 1



after feed-through addition

after iterative deletion