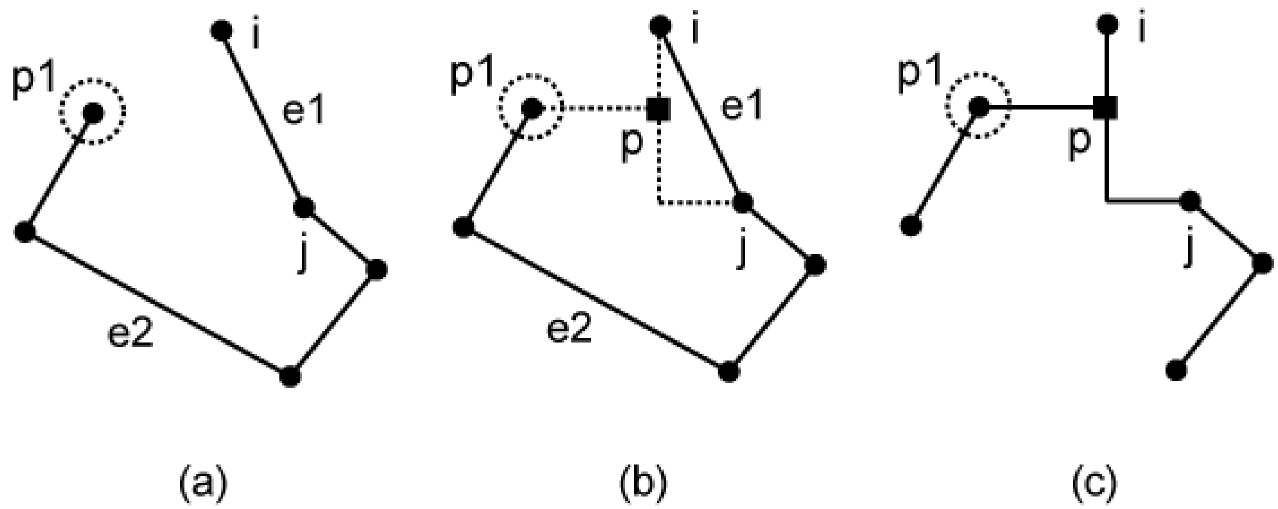
1-Steiner by Borah/Owens/Irwin

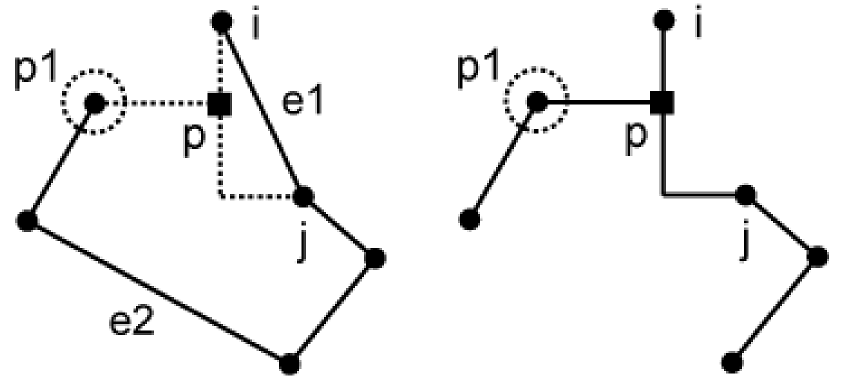
- **Interesting Observation**

  Our edge-based algorithm is based on connecting a node to the nearest point on the rectangular layout of an edge in the tree and removing the longest edge in the loop thus formed.



Gain Computation

- Things to do
  1) Add node p
  2) Remove edge e_1
  3) Remove edge e_2
  4) Add edge connecting p to p_1
  5) Add edge connecting p to p_2
  6) Add edge connecting p to p_3.



- Thus, the gain is

$$gain = length(e_2) - length(p, p_1)$$

Overall Algorithm

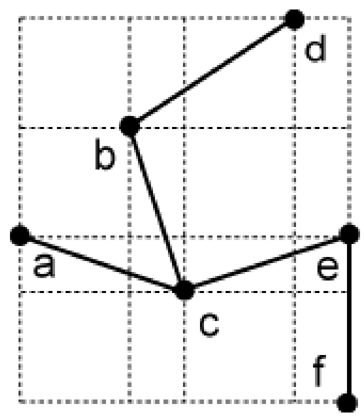
- **Multi-pass Heuristic**
  - Entire algorithm can be repeated

```
Algorithm Edge-based-Steiner()

Begin
   1.Compute the rectilinear minimum spanning tree of the set of nodes
   2.Compute all possible <node, edge> pairs that give positive gain
   3.Sort all the pairs in descending order of gain
   4.While (there are pairs with positive gain) do
            If (the two edges to be replaced exist in the tree) then
                 Replace the pair of edges with three new edges and a new node.
            End-if
   End-while
End
```

1-Steiner Routing by Borah/Owens/Irwin

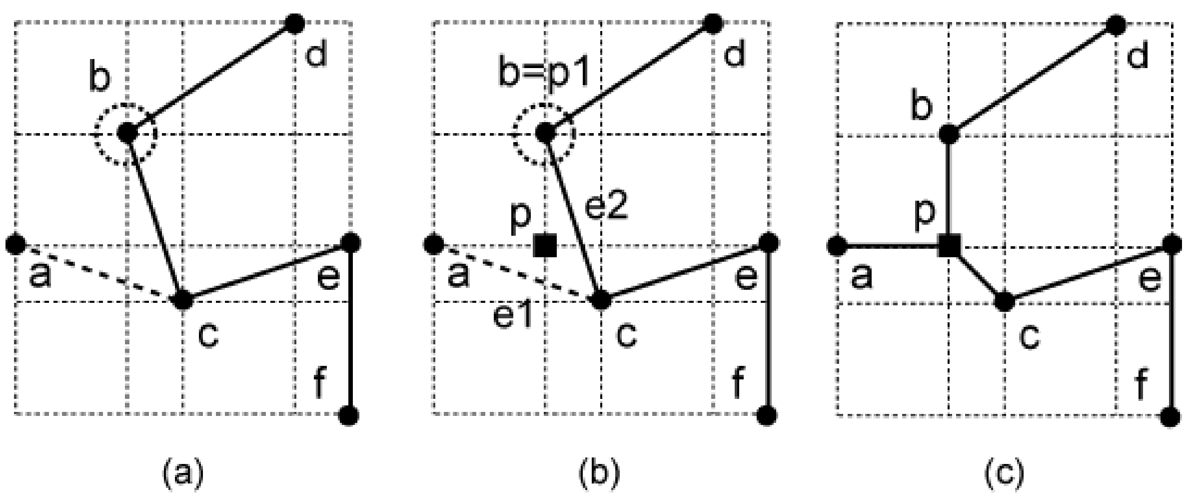
- Perform a single pass of Borah/Owens/Irwin
  - Initial MST has 5 edges with wirelength of 20
  - Need to compute the max-gain (node, edge) pair for each edge in this MST



Best Pair for (*a*,*c*)

We first let $p_1 = b$ and $e_1 = (a, c)$. Next, we compute the shortest Manhattan distance between p_1 and a "rectilinear layout" of e_1, which is 2 in this case. The node p is the nearest point on this rectilinear layout of e_1 to p_1. Next, we look for e_2, the longest edge on p_1-to-a path, which is $e_2 = (b, c)$. Thus,

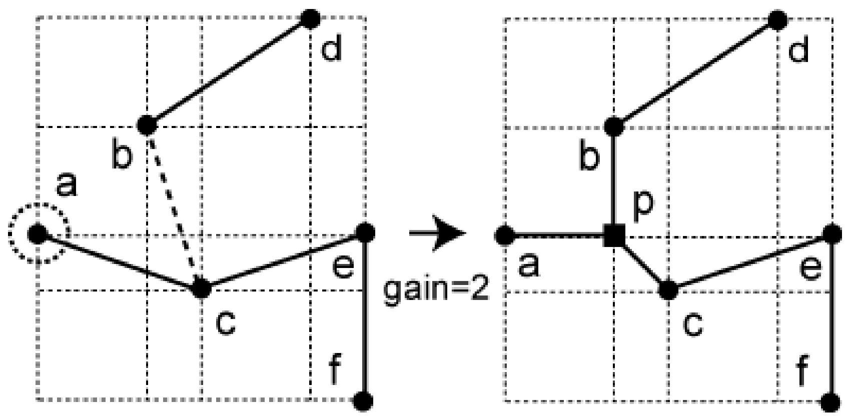
$$gain\{b, (a, c)\} = length(e_2) - length(p, p_1) = 4 - 2 = 2$$



Best Pair for (*b*,*c*)

- Three nodes can pair up with (*b*,*c*)

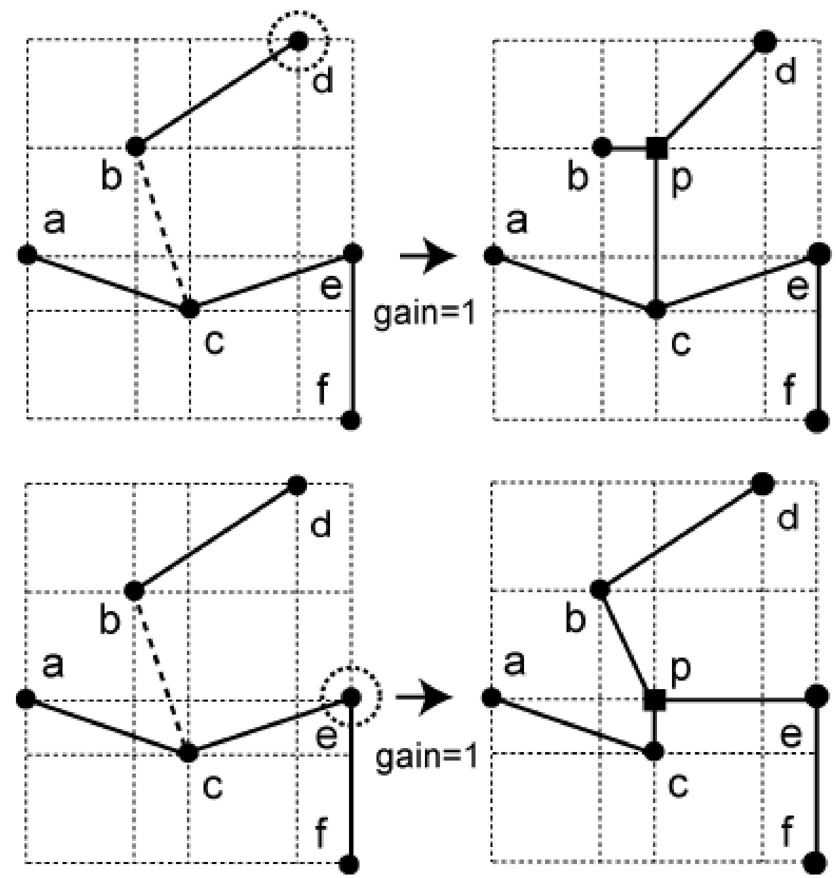
$$gain\{a,(b,c)\} = length(a,c) - length(p,a) = 4 - 2 = 2$$
$$gain\{d,(b,c)\} = length(b,d) - length(p,d) = 5 - 4 = 1$$
$$gain\{e,(b,c)\} = length(c,e) - length(p,e) = 4 - 3 = 1$$



$l(a,c) - l(p,a) = 4 - 2$

Best Pair for (*b*,*c*) (cont)

- All three pairs have the same gain
  - Break ties randomly

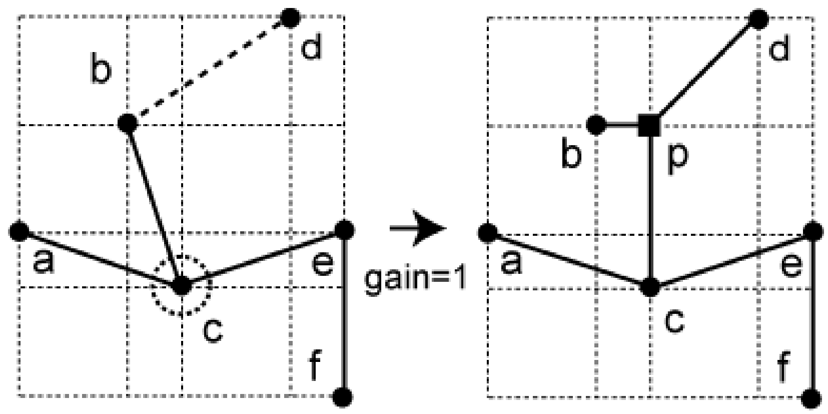


$l(b,d) - l(p,d) = 5 - 4$

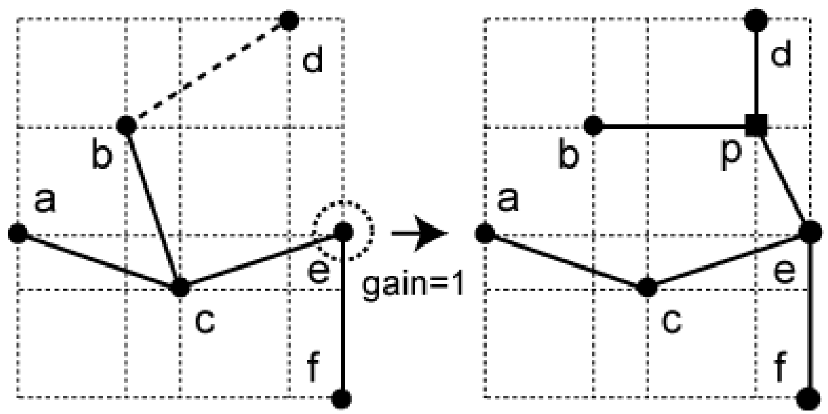
$l(c,e) - l(p,e) = 4 - 3$

Best Pair for (b,d)

- Two nodes can pair up with (b,d)
  - both pairs have the same gain



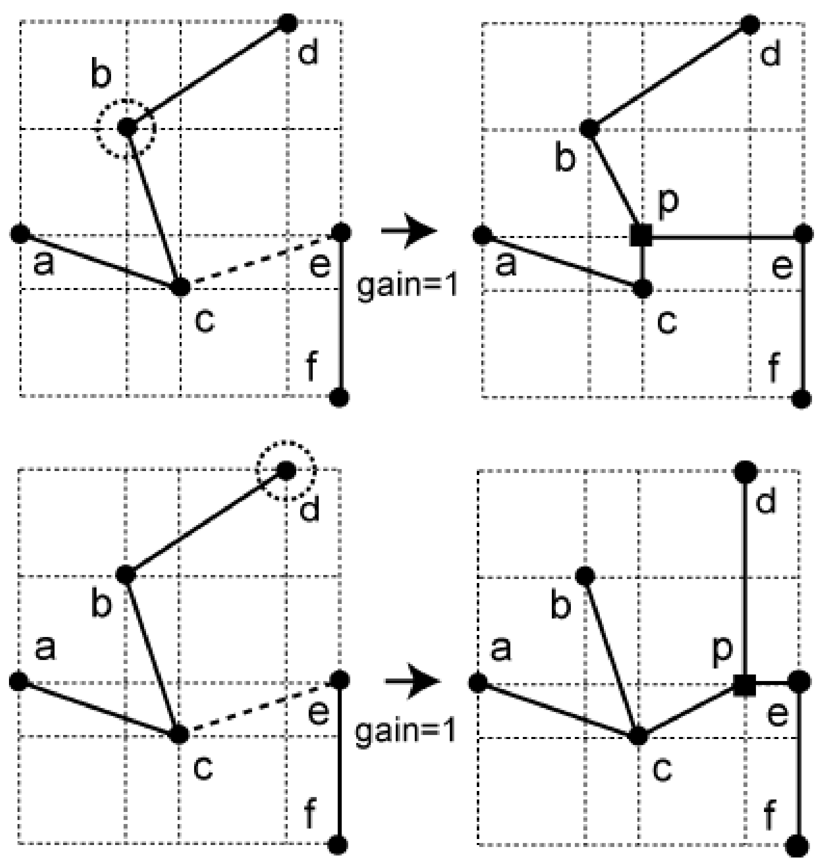
$$l(b,c) - l(p,c) = 4 - 3$$



$$l(b,c) - l(p,e) = 4 - 3$$

Best Pair for (c,e)

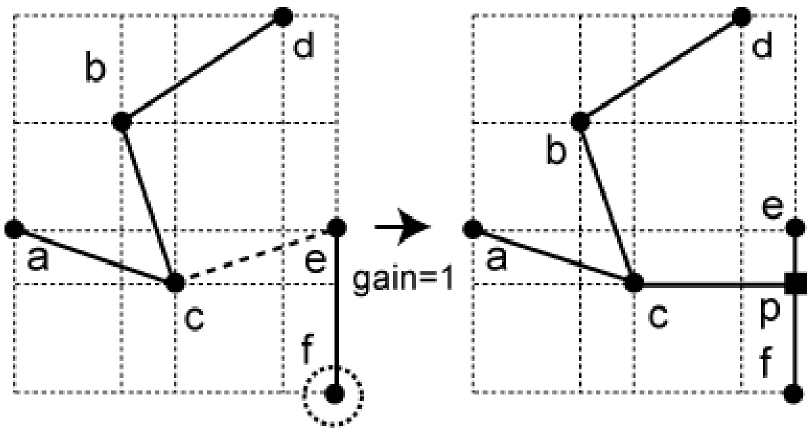
- Three nodes can pair up with (c,e)



$l(b,c) - l(p,b) = 4 - 3$

$l(b,d) - l(p,d) = 5 - 4$

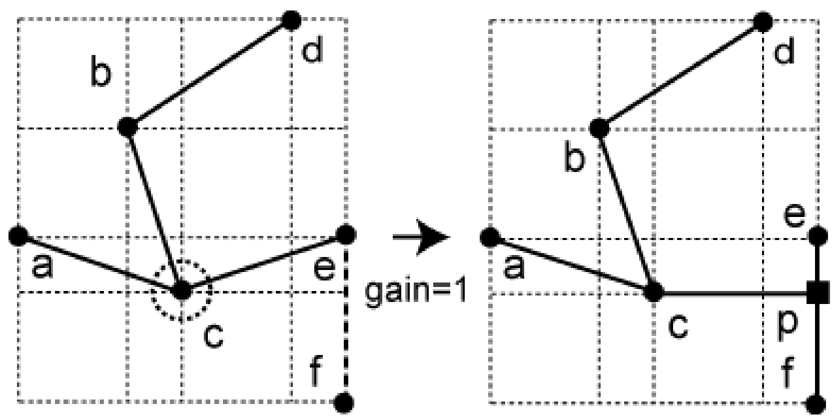
Best Pair for (*c*,*e*) (cont)



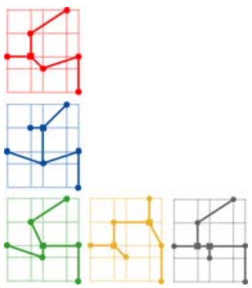
$l(e,f) - l(p,f) = 3 - 2$

Best Pair for (*e*,*f*)

- Can merge with *c* only



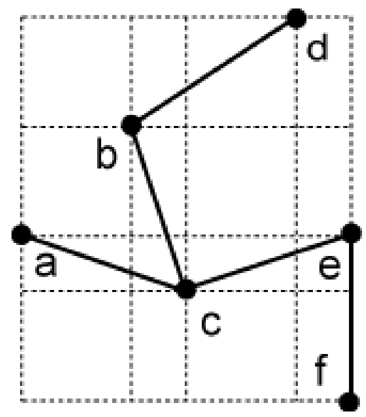
$l(c,e) - l(p,c) = 4 - 3$



Summary

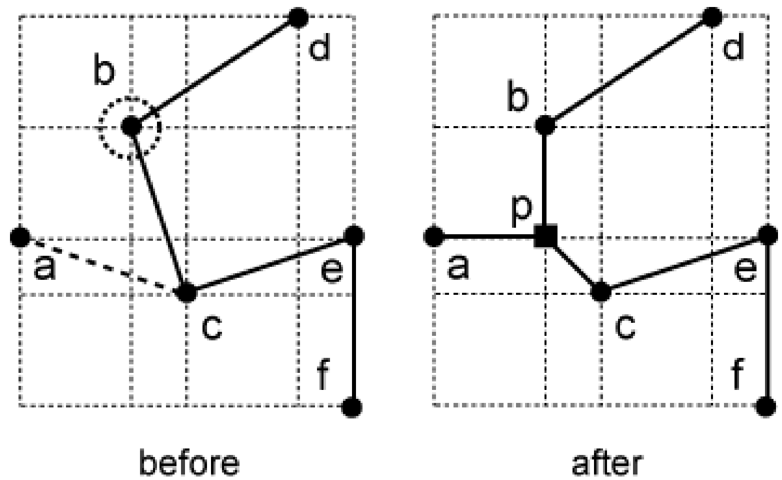
- Max-gain pair table
  - Sort based on gain value

pair	gain	e_1	e_2
$\{b, (a, c)\}$	2	(a, c)	(b, c)
$\{a, (b, c)\}$	2	(b, c)	(a, c)
$\{c, (b, d)\}$	1	(b, d)	(b, c)
$\{b, (c, e)\}$	1	(c, e)	(b, c)
$\{c, (e, f)\}$	1	(e, f)	(c, e)



First 1-Steiner Point Insertion

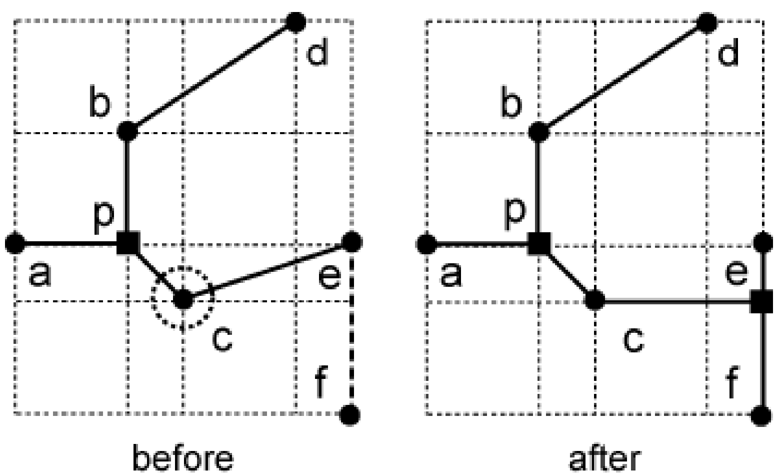
- Choose {b, (a,c)} (max-gain pair)
  - Mark $e_1 = (a,c)$, $e_2 = (b,c)$
  - Skip {a, (b,c)}, {c, (b,d)}, {b, (c,e)} since their e_1/e_2 are already marked
  - Wirelength reduces from 20 to 18



pair	gain	e_1	e_2
$\{b, (a, c)\}$	2	(a, c)	(b, c)
$\{a, (b, c)\}$	2	(b, c)	(a, c)
$\{c, (b, d)\}$	1	(b, d)	(b, c)
$\{b, (c, e)\}$	1	(c, e)	(b, c)
$\{c, (e, f)\}$	1	(e, f)	(c, e)

Second 1-Steiner Point Insertion

- Choose {c, (e,f)} (last one remaining)
  - Wirelength reduces from 18 to 17



pair	gain	e_1	e_2
$\{b, (a, c)\}$	2	(a, c)	(b, c)
$\{a, (b, c)\}$	2	(b, c)	(a, c)
$\{c, (b, d)\}$	1	(b, d)	(b, c)
$\{b, (c, e)\}$	1	(c, e)	(b, c)
$\{c, (e, f)\}$	1	(e, f)	(c, e)