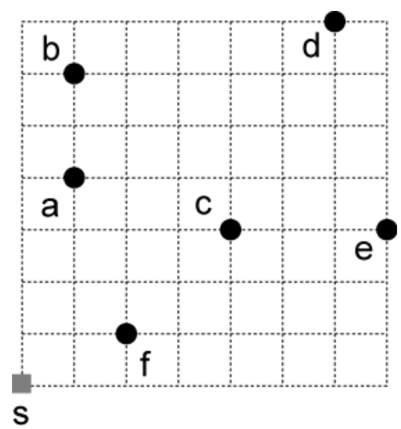
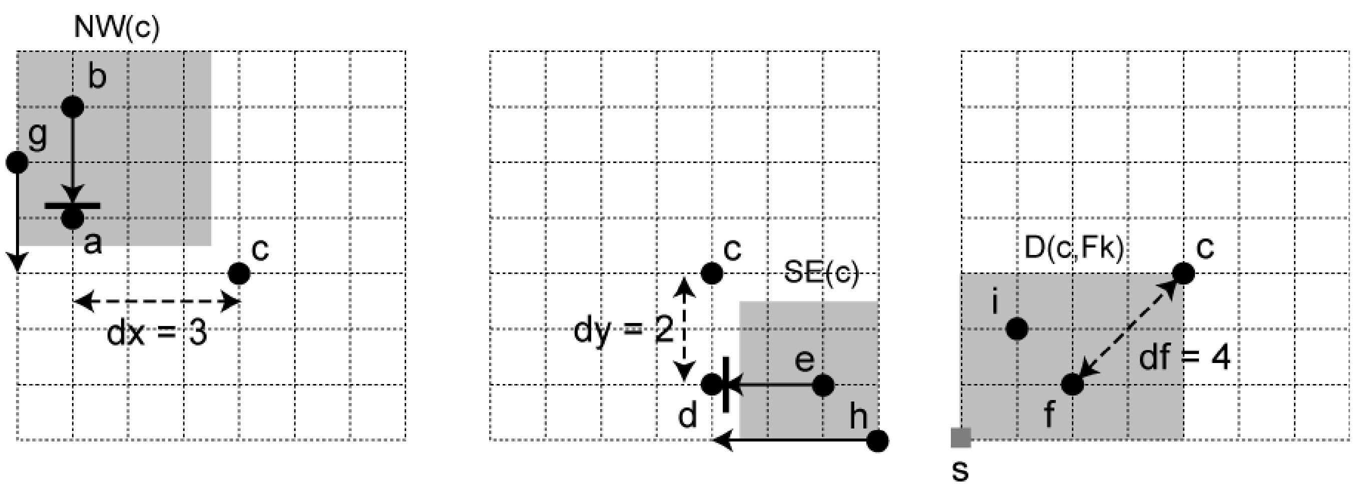
A-tree Routing Algorithm

- Compute $dx(c, F_0)$, $dy(c, F_0)$, $df(c, F_0)$
  - We begin with root set $R(F_0) = \{a,b,c,d,e,f\}$ for initial forest F_0



Recall that …

- $dx(p, F)$: we first compute the set of root nodes located in the northwest of p that are not blocked from p. From this set, we choose $q = mx(p, F_k)$ with the minimum horizontal distance $d_H(p, q)$. $dx(p, F_k)$ is this minimum $d_H(p, q)$ value. See Figure (a). ***mx = a, dx = 3***
- $dy(p, F_k)$: we first compute the set of root nodes located in the southeast of p that are not blocked from p. From this set, we choose $q = my(p, F_k)$ with the minimum vertical distance $d_V(p, q)$. $dy(p, F_k)$ is this minimum $d_V(p, q)$ value. See Figure (b). ***my = d, dx = 2***

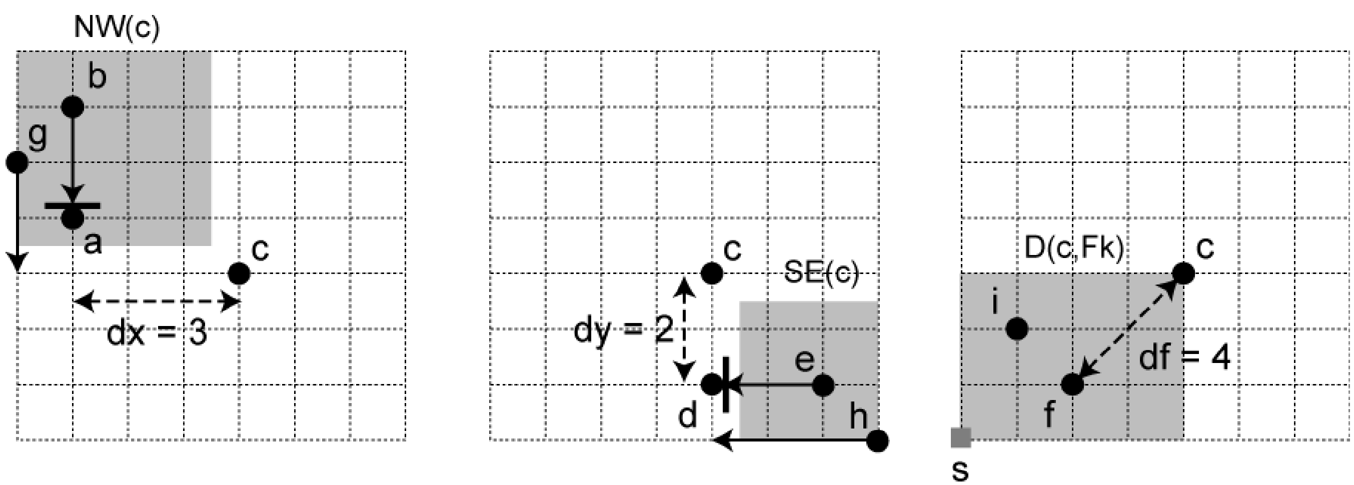


(a) (b) (c)

Recall that … (cont)

- $df(p, F_k)$: we first compute $MF(p, F_k)$, the set of nodes (= not necessarily root nodes) that are dominated by p and are separated by p with the minimum rectilinear distance. $df(p, F_k)$ is this minimum rectilinear distance value. In addition, we compute mk_w, the node in $MF(p, F_k)$ with the minimum x-coordinate. Similarly, mk_s is the node in $MF(p, F_k)$ with the minimum y-coordinate. See Figure (c).

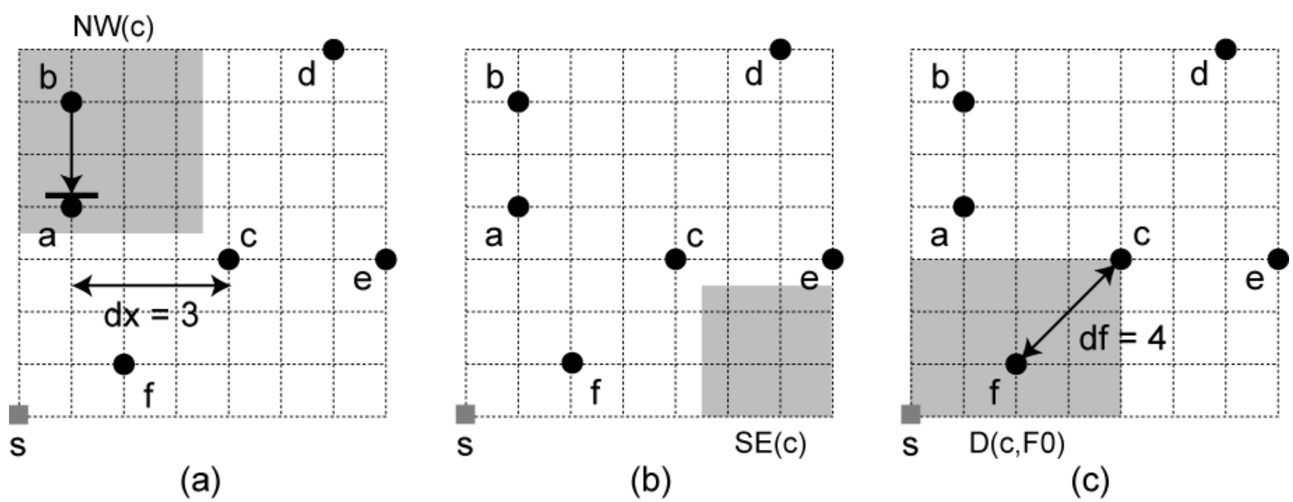
MF = {f, i}, df = 4, mf_w = i, mf_s = f



(a) (b) (c)

Computing dx/dy/df for Node *c*

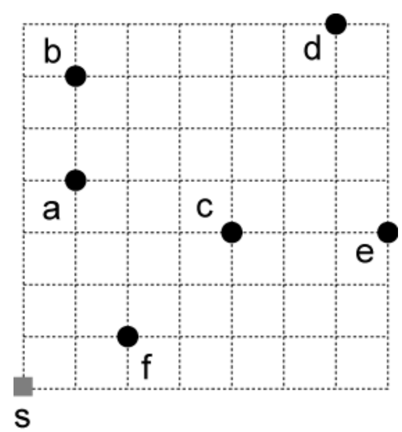
- $dx(c, F_0)$: we see that $NW(c) \cap R(F_0) = \{a, b\}$ as shown in Figure (a). In this case, node b is blocked from node c (= by a) while a is not. Thus, we have $mx(c, F_0) = a$. Since $d_H(a, c) = 3$, we have $dx(c, F_0) = 3$.
- $dy(c, F_0)$: we see that $SE(c) \cap R(F_0) = \emptyset$ as shown in Figure (b). Thus, we have $my(c, F_0) = \emptyset$, and $dy(c, F_0) = \infty$.
- $df(c, F_0)$: we see that $D(c, F_0) = \{s, f\}$ as shown in Figure (c). Thus, we have $MF(c, F_0) = \{f\}$ and $df(c, F_0) = 4$. Since f is only node in $MF(c, F_0)$, we have $mf_w(c, F_0) = mf_s(c, F_0) = f$.



Computing dx/dy/df Values

- Compute *dx/dy/df* for all other nodes

p	mx	dx	my	dy	MF	mf_w	mf_s	df
a	$\emptyset$	∞	c	1	$\{s\}$	s	s	5
b	$\emptyset$	∞	c	3	$\{a\}$	a	a	2
c	a	3	$\emptyset$	∞	$\{f\}$	f	f	4
d	$\emptyset$	∞	e	4	$\{b, c\}$	b	c	6
e	d	1	$\emptyset$	∞	$\{c\}$	c	c	3
f	a	1	$\emptyset$	∞	$\{s\}$	s	s	3



Safe Move Computation

- What kind of safe moves does node *a* contain?
  - We have $dx(a, F_0) = \infty$, $dy(a, F_0) = 1$, $df(a, F_0) = 5$
    - Type 1: $dx \geq df$ and $dy \geq df$
    - Type 2: $dx \geq df$ and $dy < df$
    - Type 3: $dx < df$ and $dy \geq df$
  - So *a* has type-2 safe move

Safe Move Computation (cont)

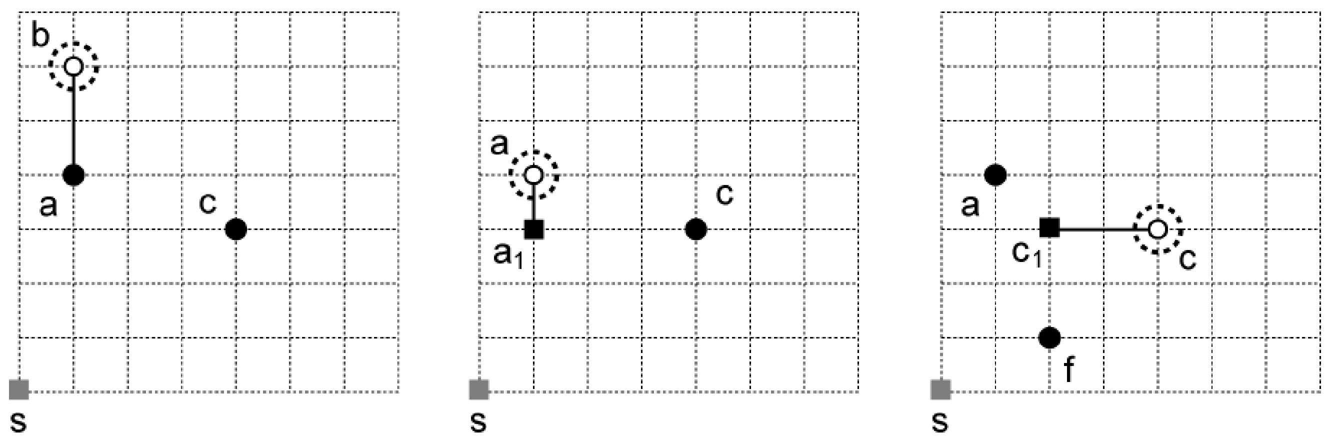
- Compute safe moves for all nodes in F_0
  - Type 1: $dx \geq df$ and $dy \geq df$
  - Type 2: $dx \geq df$ and $dy < df$
  - Type 3: $dx < df$ and $dy \geq df$
  - All moves are safe
    - No heuristic moves necessary

node	type-1	type-2	type-3
a	no	yes	no
b	yes	no	no
c	no	no	yes
d	no	yes	no
e	no	no	yes
f	no	no	yes

p	mx	dx	my	dy	MF	mf_w	mf_s	df
a	$\emptyset$	∞	c	1	$\{s\}$	s	s	5
b	$\emptyset$	∞	c	3	$\{a\}$	a	a	2
c	a	3	$\emptyset$	∞	$\{f\}$	f	f	4
d	$\emptyset$	∞	e	4	$\{b, c\}$	b	c	6
e	d	1	$\emptyset$	∞	$\{c\}$	c	c	3
f	a	1	$\emptyset$	∞	$\{s\}$	s	s	3

Recall that …

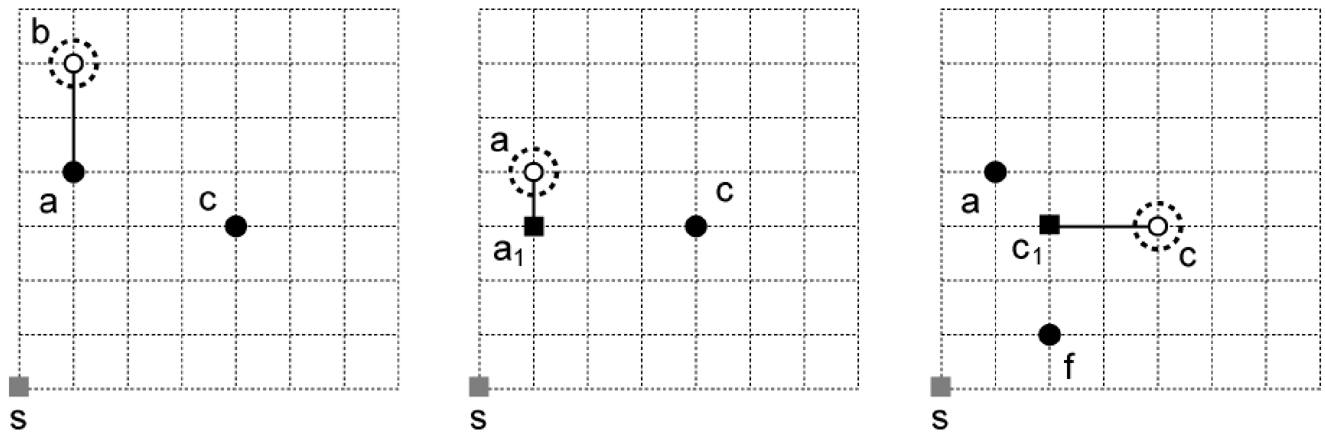
- Type-1: we add a path that connects p to mf_w. We remove p from $R(F_k)$. This move merges two trees. See Figure (a).
- Type-2: we add a down-ward vertical path of length p' from p, where p' is the minimum between (1) the vertical distance between p and $mf_s(p, F_k)$, and (2) $dy(p, F_k)$. We remove p from $R(F_k)$ and add p'. This move grows the tree rooted at p. See Figure (b).



(a) (b) (c)

Recall that … (cont)

- Type-3: we add a left-ward horizontal path of length p' from p, where p' is the minimum between (1) the horizontal distance between p and $mf_w(p, F_k)$, and (2) $dx(p, F_k)$. We remove p from $R(F_k)$ and add p'. This move grows the tree rooted at p. See Figure (c).

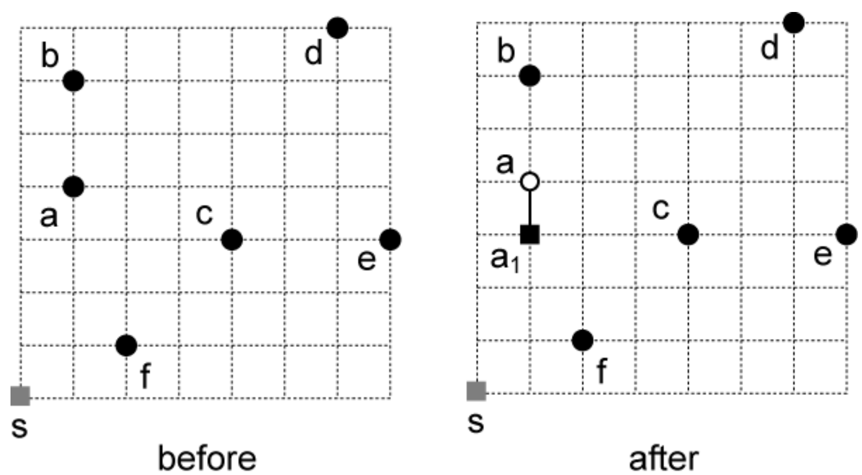


(a) (b) (c)

Safe Move for Node *a*

- Perform safe move for node *a* (type 2)

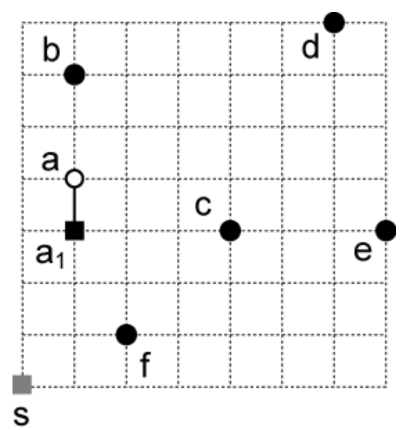
  We see that $d_V(mf_s(a, F_0), a) = d_V(s, a) = 4$, and $dy(a, F_0) = 1$. Thus, the length of vertical path to be added to node a is $\min\{4, 1\} = 1$. We connect a to a newly added root node a_1. We then update $R(F_1) = R(F_0) - \{a\} + \{a_1\} = \{a_1, b, c, d, e, f\}$.



Safe Move for Node *a* (cont)

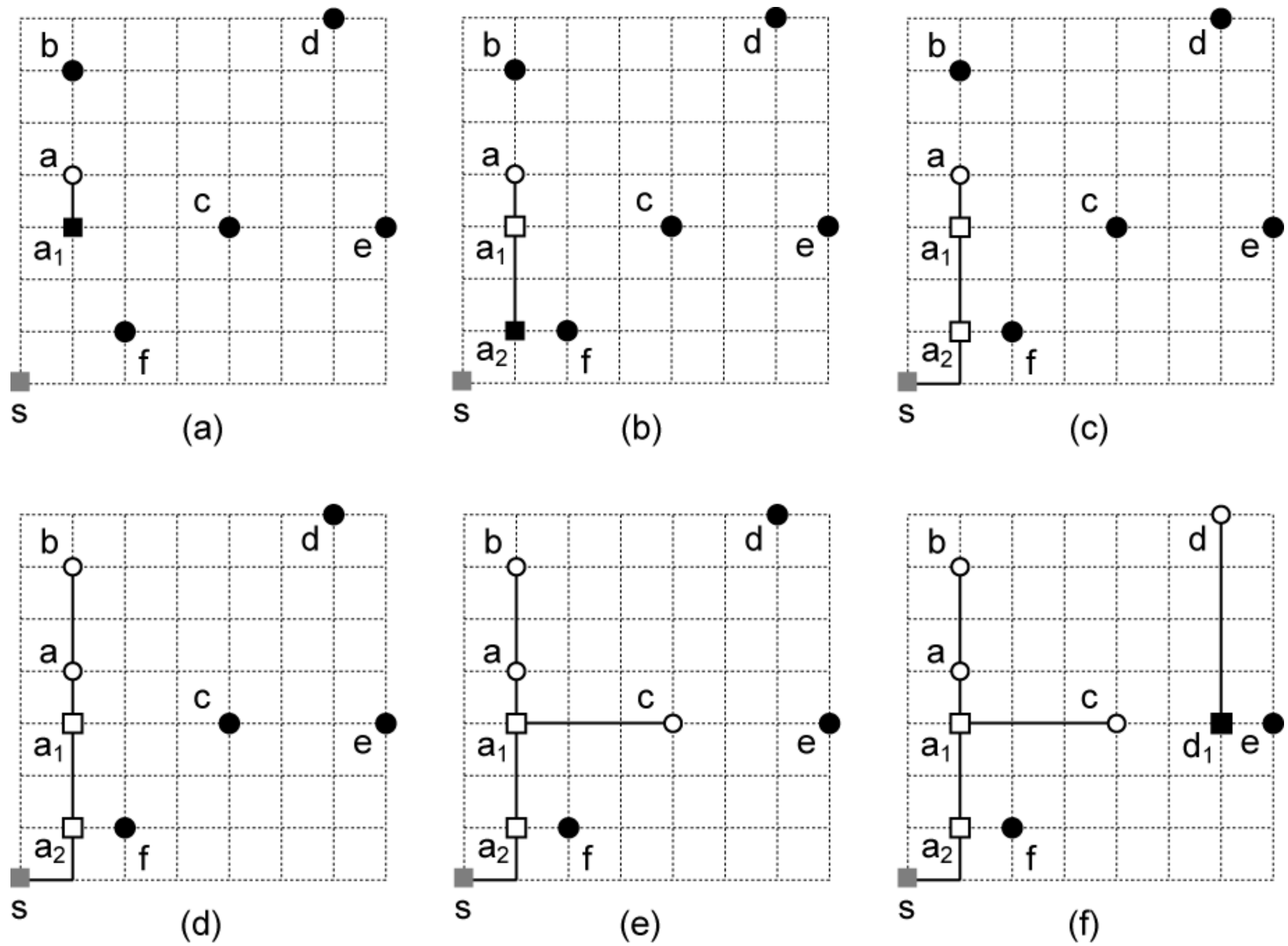
- Updating *dx/dy/df* values and safe moves

p	mx	dx	my	dy	MF	mf_w	mf_s	df	type-1	type-2	type-3
a_1	$\emptyset$	∞	f	2	$\{s\}$	s	s	4	no	yes	no
b	$\emptyset$	∞	c	3	$\{a\}$	a	a	2	yes	no	no
c	$\emptyset$	∞	$\emptyset$	∞	$\{a_1\}$	a_1	a_1	3	yes	no	no
d	$\emptyset$	∞	e	4	$\{b, c\}$	b	c	6	no	yes	no
e	d	1	$\emptyset$	∞	$\{c\}$	c	c	3	no	no	yes
f	a_1	1	$\emptyset$	∞	$\{s\}$	s	s	3	no	no	yes



Performing Remaining Safe Moves

- Choose the nodes in alphabetical order



Performing Remaining Moves

- Final rectilinear Steiner arborescence
  - All source-sink paths are shortest
  - Total wirelength = 18
  - 3 Steiner nodes (white square) used
  - All moves performed were safe