

state s.

$\langle L-O-S-K \rangle$

Neighbours:

$\langle O-L-S-K \rangle$
1.7

$\langle S-O-L-K \rangle$
1.8

$\langle K-O-S-L \rangle$
1.3

$\langle L-S-O-K \rangle$
1.3

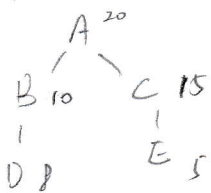
$\langle L-K-S-O \rangle$
1.2

$\langle L-O-K-S \rangle$
1.5

(i) As we can see above, $\langle L-K-S-O \rangle$, which is one of initial state's neighbour, has lower cost of path $1.2 < 1.8$. Therefore next step should choose the state $\langle L-K-S-O \rangle$.

(ii) We stop until no neighbour of current state has a lower value cost. If there exist a neighbour that has lower cost path, we change the state into the neighbour one.

(iii) We cannot ensure our solution is global optimal. Given the example:



Suppose we are to find minimum value of the tree. When we at state A we compare the value B and C. And choose to continue search through B as $B < C$. We ignore the C state, but actually its child E has lowest value of all. We can only get $D=8$ as local minimal.