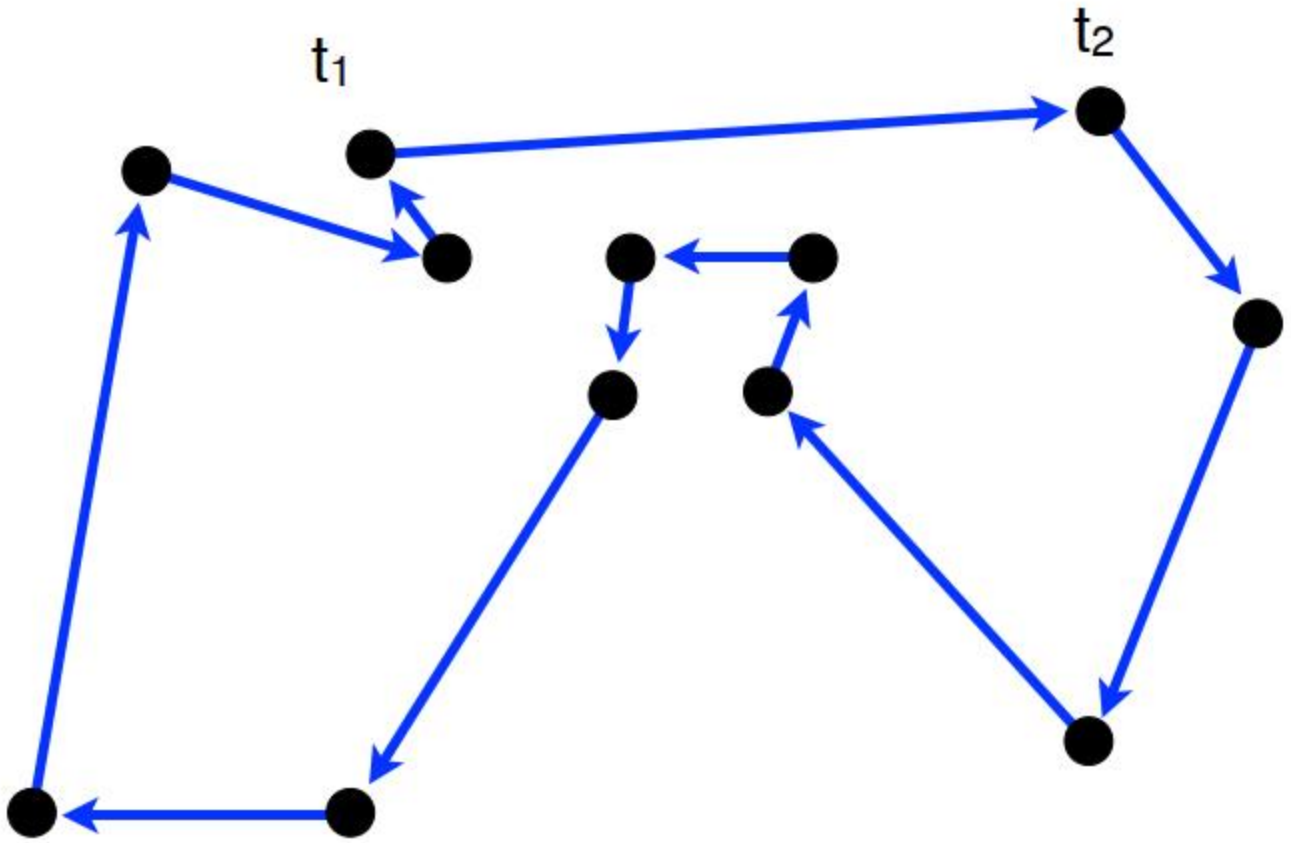
K-OPT (first iteration)

Choose a vertex t1.

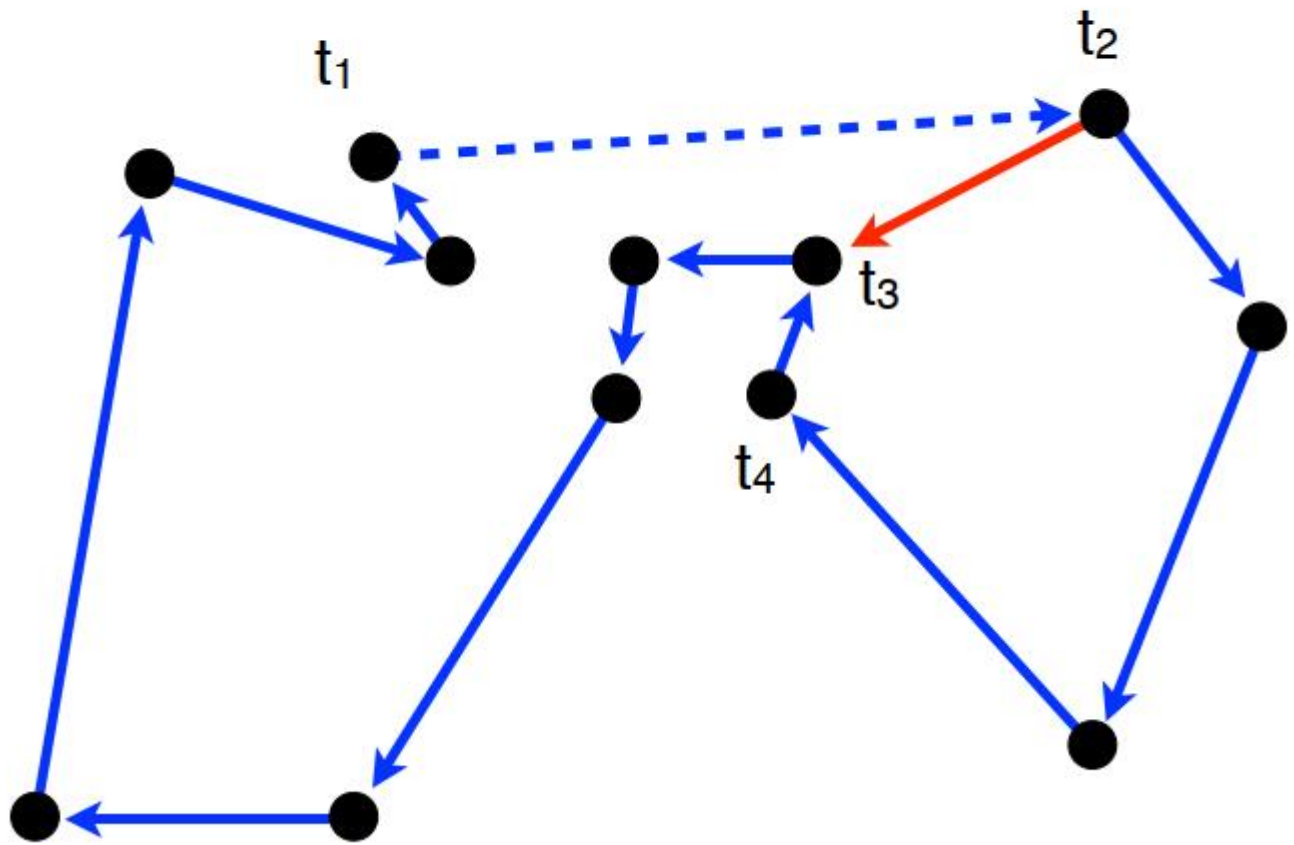
It's currently connected along edge (t1,t2)



K-OPT (first iteration)

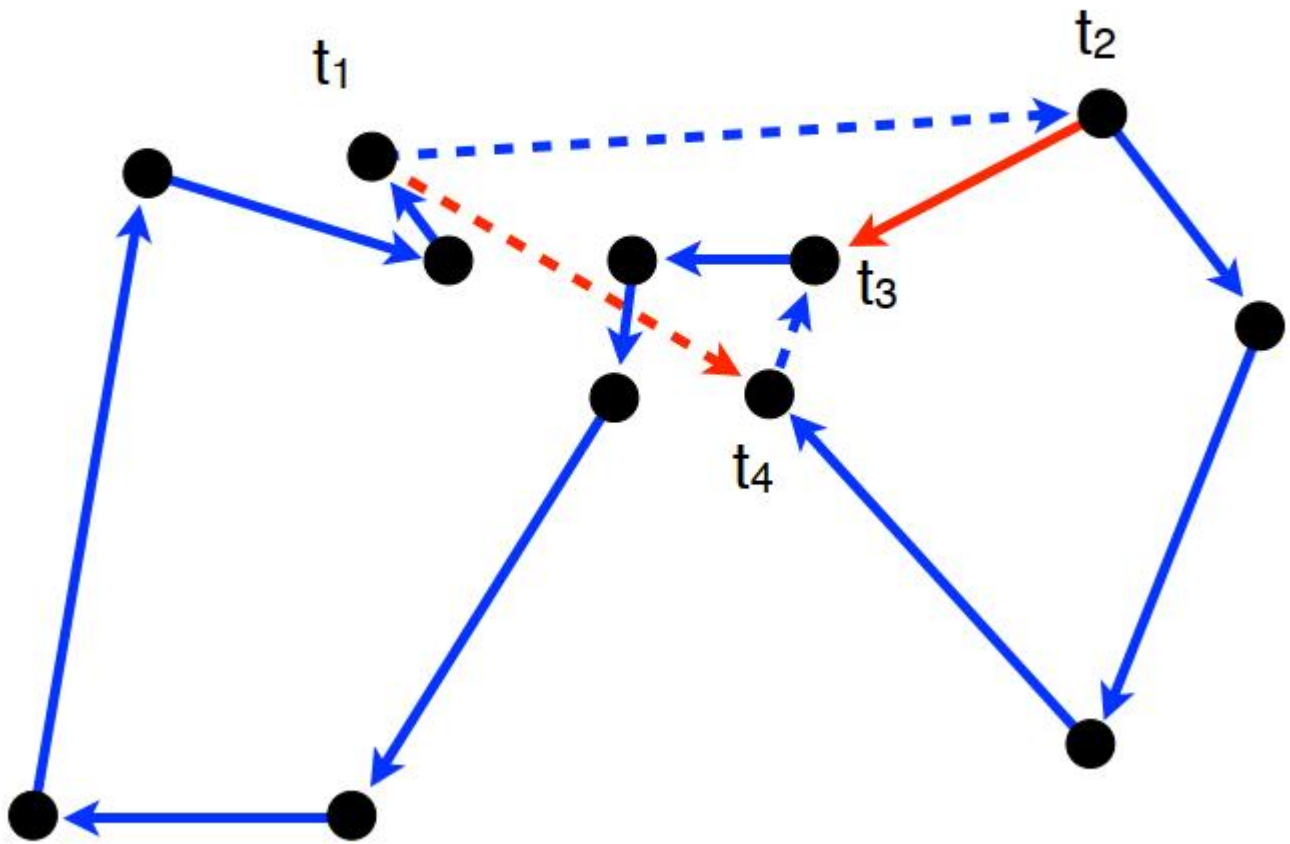
Find a vertex t3, so that the edge (t2,t3)
..which is not in the tour yet…

Is < the edge (t1,t2)



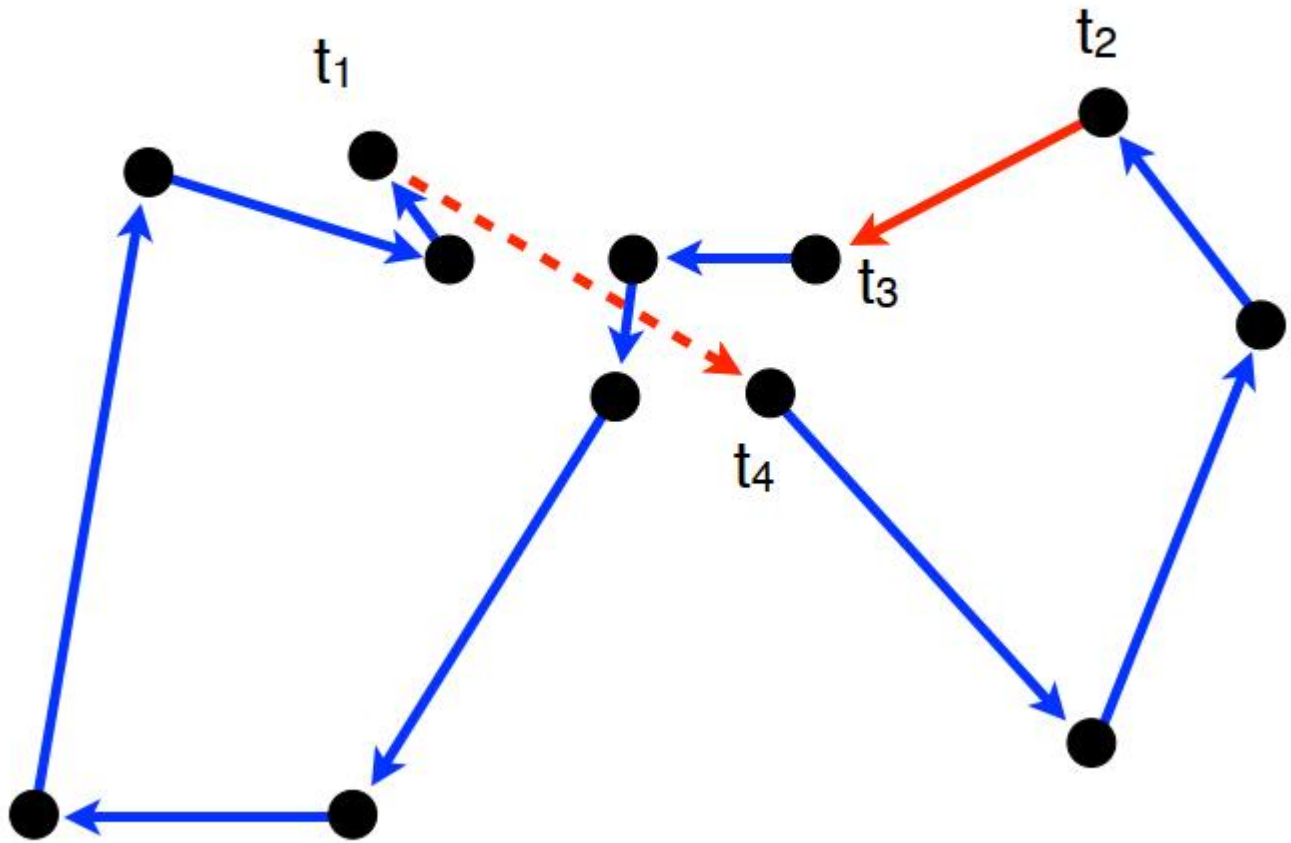
K-OPT (first iteration)

The t3 vertex was reached from t4 in the tour, so now it connects to t1



K-OPT (first iteration)

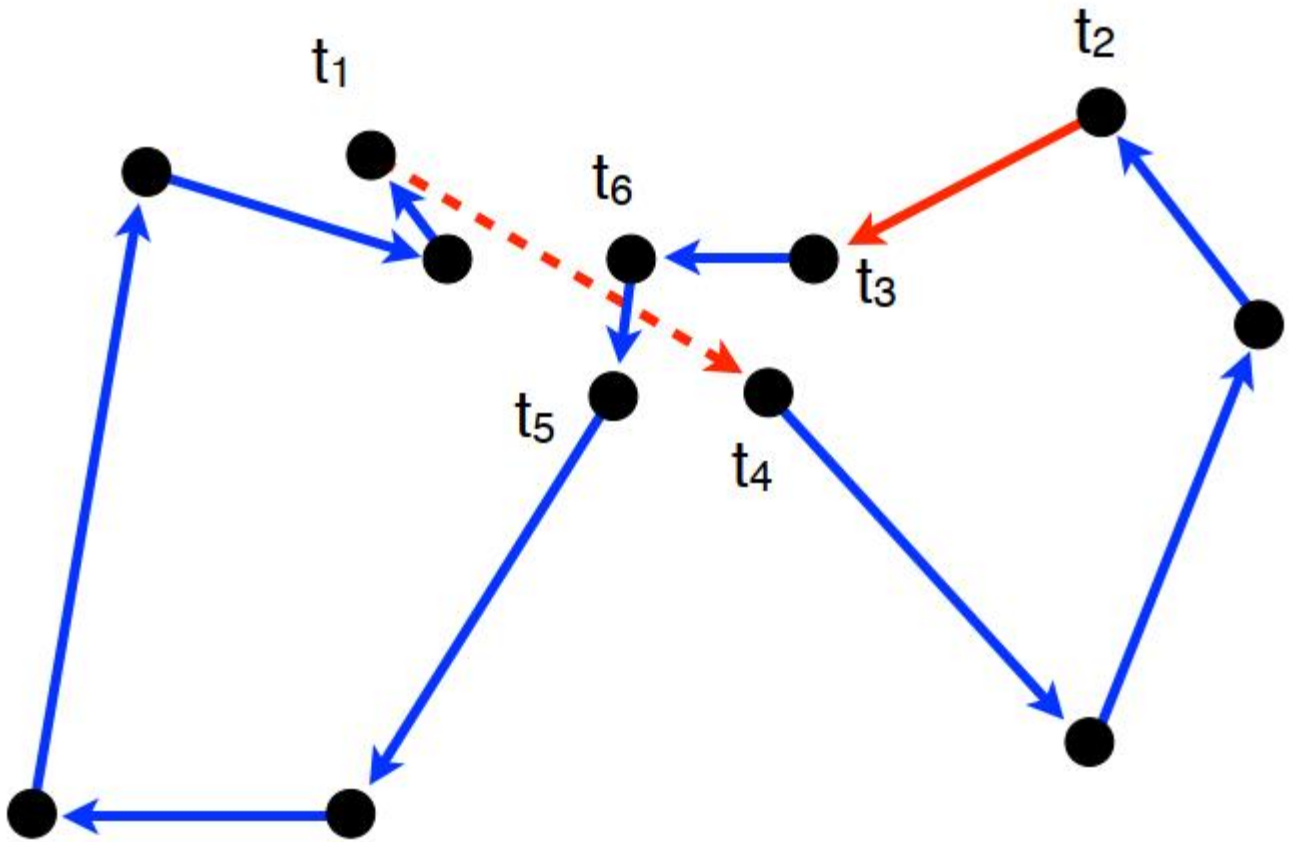
The tour after the first edge swap (this is like a 2-opt move)



K-OPT (first iteration)

t1 is connected to t4 now.

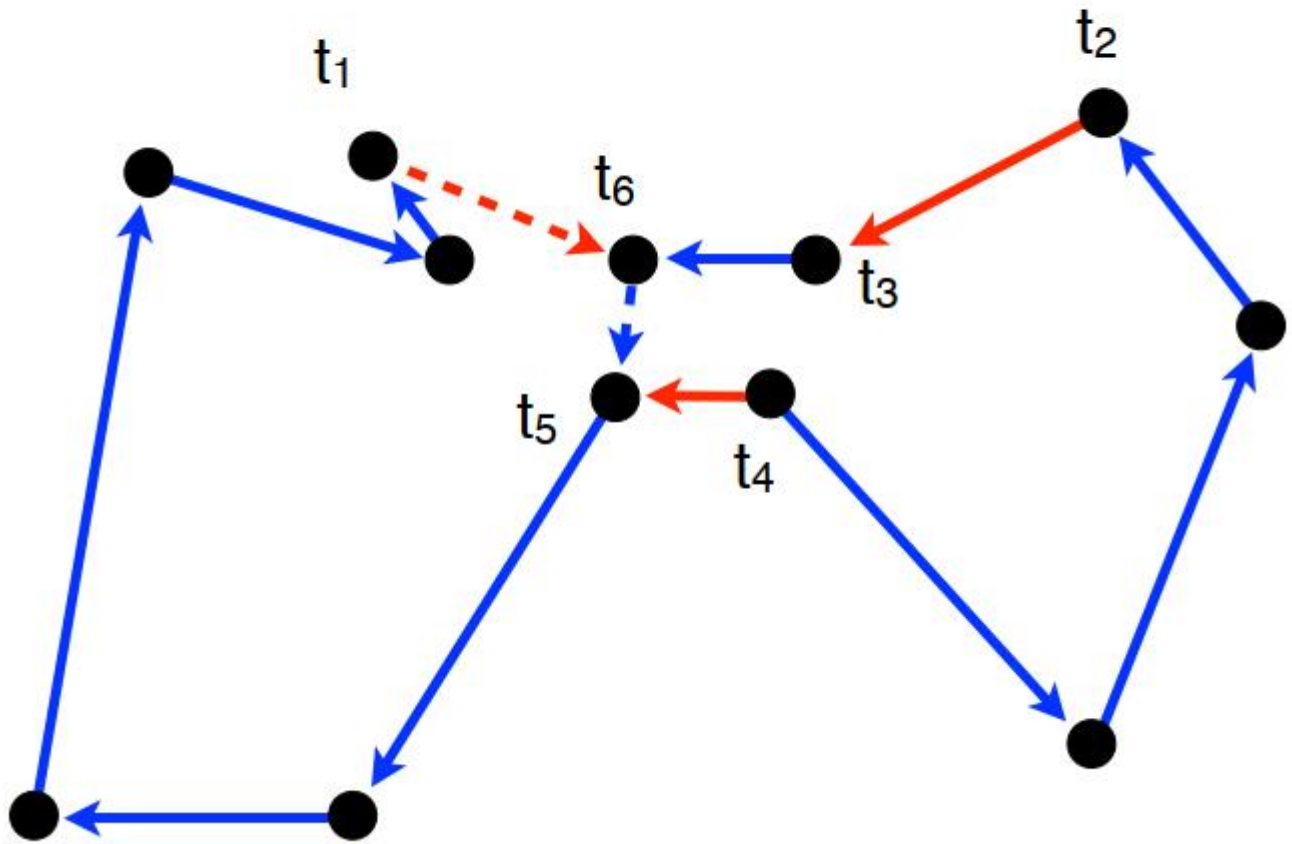
Find a vertex t5 where (t4,t5) is < (t1,t4)



K-OPT (first iteration)

Replace the (t1,t4) edge with (t4,t5)

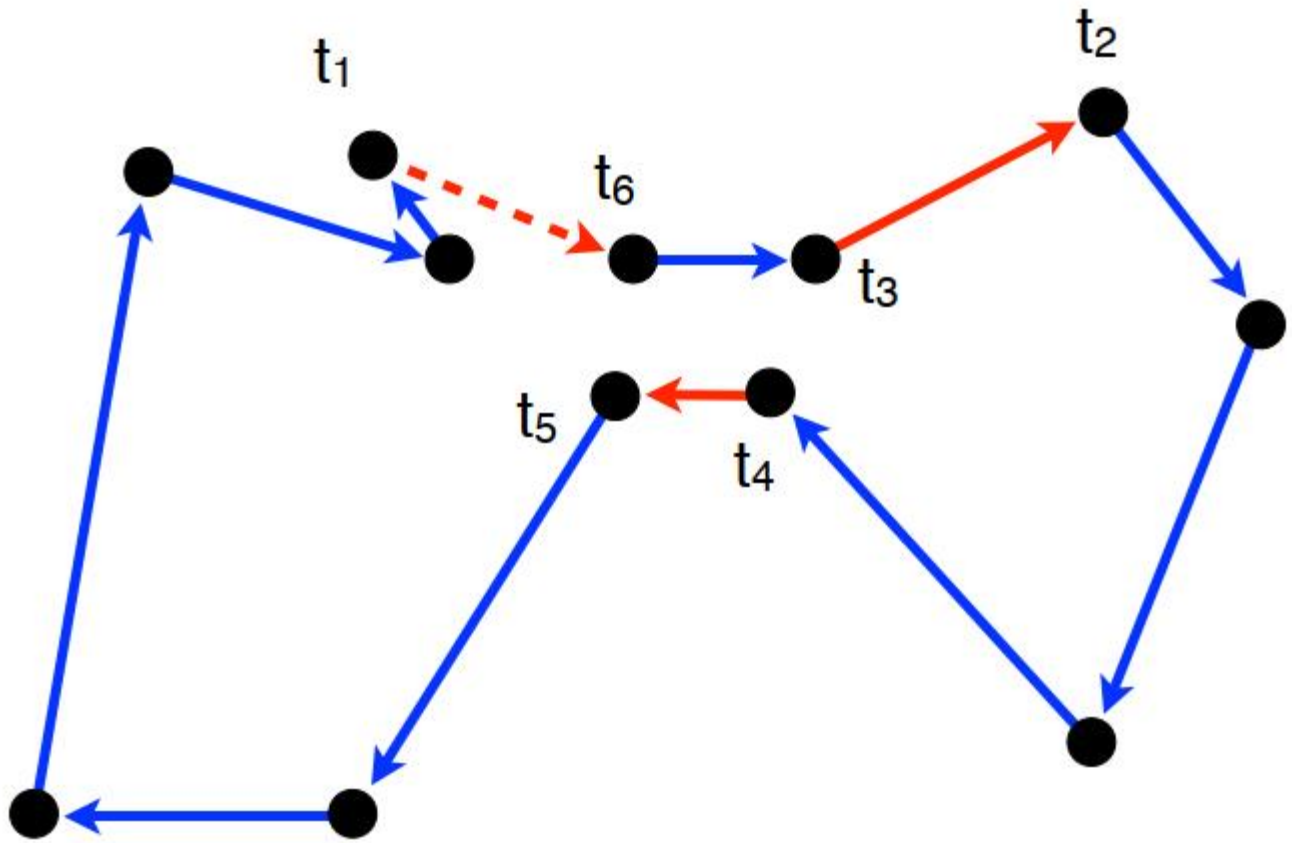
The t5 vertex was connected to t6, so now make t6 connect to t1



K-OPT (first iteration)

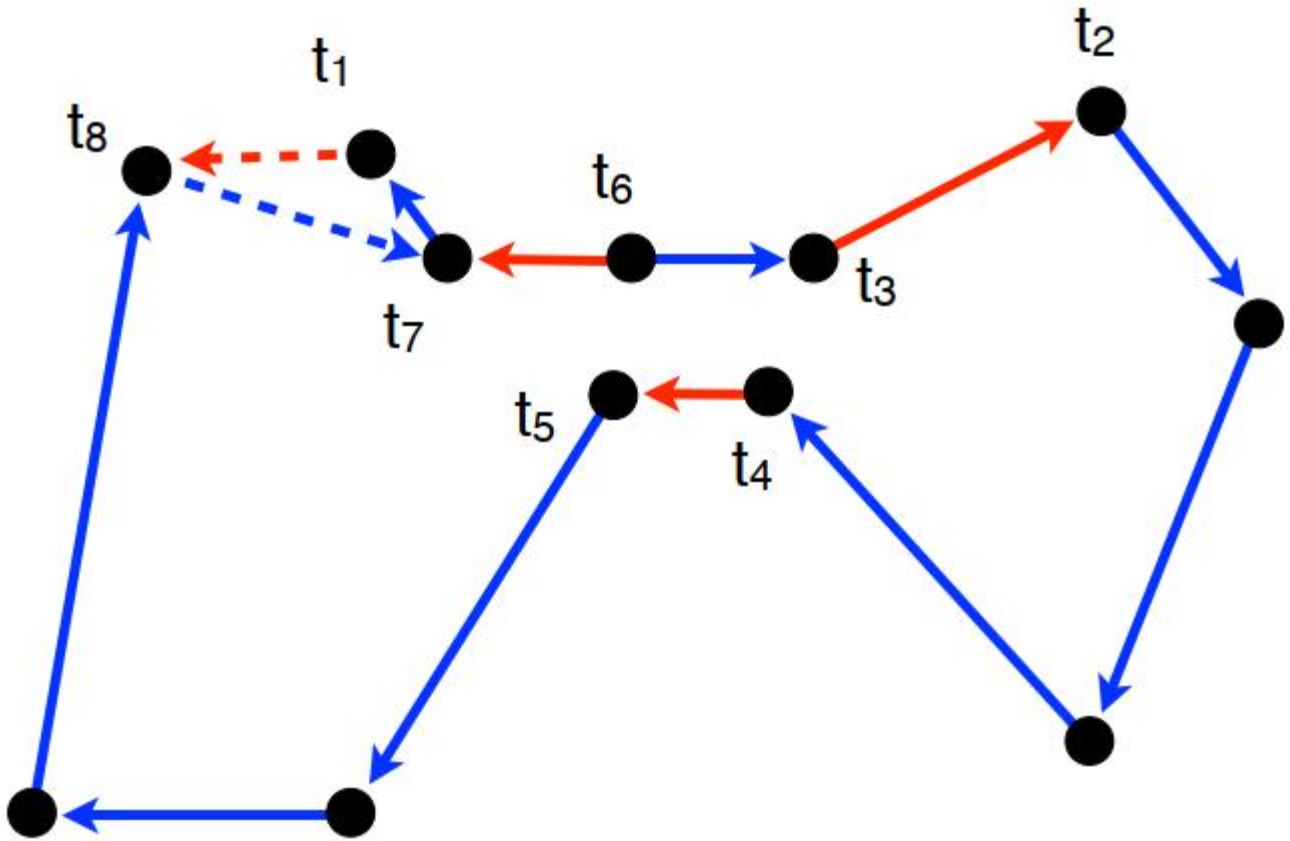
We've swapped 3 edges around
(Sort of like a 3-OPT move)

Now choose an edge (t6,t7)
Whose cost is < (t1,t6)



K-OPT (first iteration)

Replace (t1,t6) and (t7,t8) with (t6,t7) and (t1,t8)

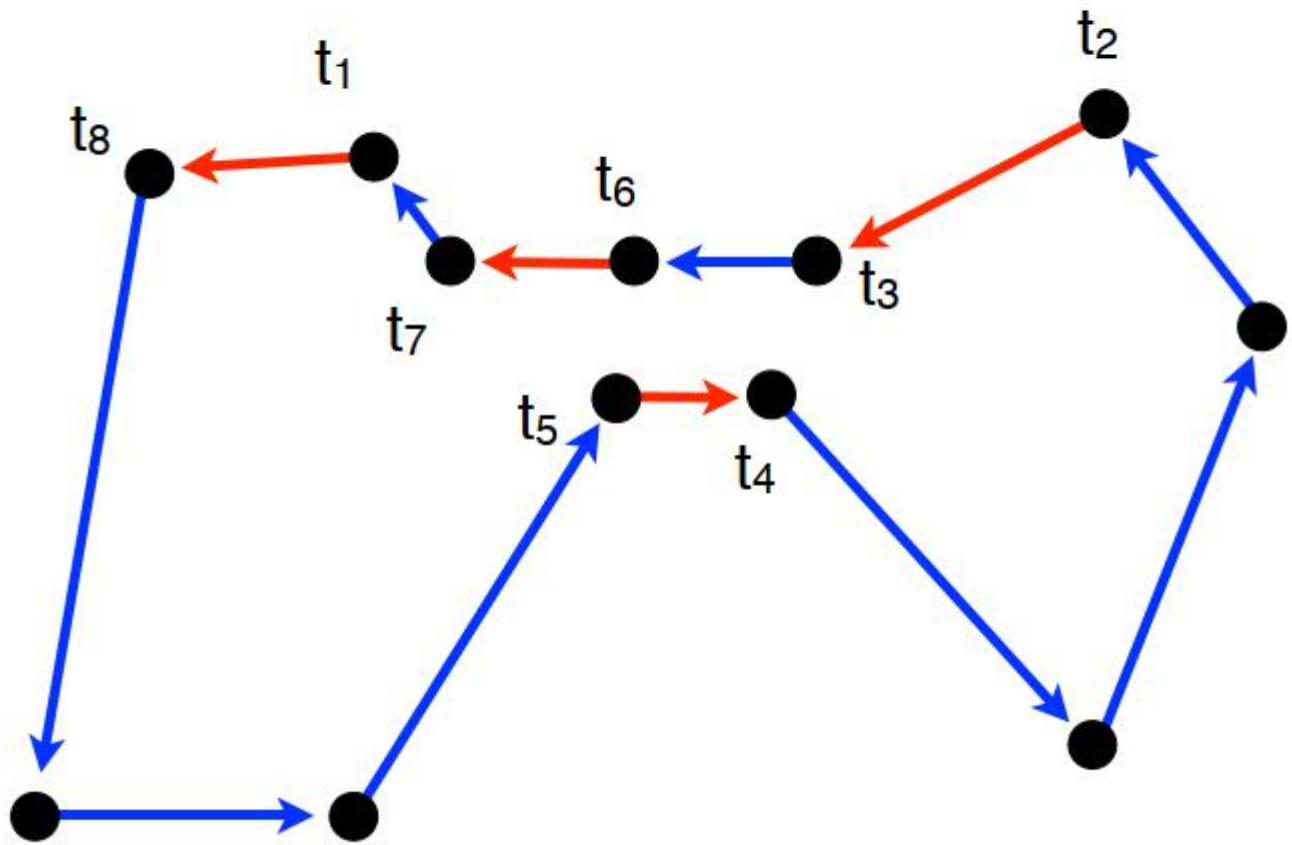


K-OPT (first iteration)

Keep on going until you can't find a new edge that is shorter.

Out of all of the swaps you did, Remember the tour with the best cost and keep it.

That's one change.



K-OPT (second iteration)

The second iteration starts this Process all over again, with a new starting t1 vertex.