**multi-class improved Traffic Assignment by Paired Alternative Segments (m-iTAPAS)**

*Input*:

: Minimal flow level

: Minimal cost level

Tolerance level

Step 1. Initialize origin-based arc flows , origin-based reduced arc cost , least cost path predecessor labels for every origin, every vehicle class , and an empty set for paired alternative segments

Step 2. Perform AON assignment – From each origin find the least cost path to every destination for every vehicle class and assign all demand on arcs of this path

for

for

for

for end

end

end

end

Step 3. Iterate to shift flows between arcs until the algorithm converges

while

for

for

Step 3.1 Identify arcs with substantial flow and substantial reduced cost

for

for

If and and

Step 3.2. Develop Paired Alternative Segment (PAS) for the potential arc

using Maximum Cost Search (MCS) algorithm and perform flow shift

If end

end

end

end

end

Step 3.3. Perform flow shifts on all identified PAS to fasten the algorithm convergence

for

end

end

Step 4. Remove PAS which no longer results in significant improvement in the solution

for

for

if or and

else

end

end

end

Step 5. If the relative gap is smaller than the tolerance level then the algorithm is said to have

converged

if

end

end

return

**PAS identification using Maximum Cost Search**

*Inputs*:

: Arc as

: origin node

Step 1. Initialize status label and predecessor label for each node

Step 2. Set the tail and head node on arc and iterate

while

Step 2.1. Set the head node to the tail and set the tail node to the node with maximum cost arc

to the head node

Step 2.2. Set the tail node as the predecessor label for the head node

Step 2.3. If this predecessor is on the least cost path from origin to node , then a PAS can be

established

if

Step 2.3.1. Establish the first segment of the PAS using least cost labels

Step 2.3.2. Establish the second segment of the PAS using maximum cost labels

break

Step 2.4. If this predecessor is a previously identified predecessor, then a loop has been

established and a PAS cannot be established

elseif

break

Step 2.5. Else update the status of this predecessor and continue to step 2.1

else

end

end

return

**Flow Shift on PAS inspired by Newton Method inherited from Dial (2006)**

*Inputs*:

: Paired alternative segments (PAS)

: Vehicle class associated with PAS

: Origin associated with PAS

if

elseif

else

for end

for end

**Dijkstra’s labeling algorithm**

*Input*:

: Origin

: Destination

Step 1. Initialize a set of open nodes , cost label and predecessor label for each node

Step 2. Develop Dijkstra’s labels

Step 2.1. Set origin as the pivot node, origin cost label to zero, origin predecessor label to itself, and remove the pivot node from the set of open nodes

while

Step 2.2 Update cost and predecessor label for every neighboring node from the pivot node

 for

if

end

end

Step 2.3. From the set of open nodes, set the node with the smallest cost label as the pivot

node, and remove it from the set of open nodes

end

return

**Developing least cost path from Dijkstra’s labels**

*Input*:

:Predecessor labels

: Origin

: Destination

Step 1. Start from destination and visit predecessors using the predecessor labels until arriving at the origin

while

end

Step 2. Revert the sequence of nodes followed from destination to origin

reverse

return

**Developing least cost tree from Dijkstra’s labels**

*Input*:

:Predecessor labels

: Origin

Step 1. Initialize an empty set

Step 2. Connect every node to its predecessor node using the predecessor label to develop the least cost tree rooted at origin node

for

if and

end

end

return