

PDP – Pseudo Code

Heuristics Function: -

Given a route 'r'

Repeat:

For each pair of locations in 'r':

If (latter location is more urgent WRT time window), then:

Swap the current 2 nodes to get a route r'

Calculate Diff = Cost(r')-cost(r)

If Diff < 0 (i.e., new route is cheaper):

r=r'

Until: no better routes available (Tabu Search)

Cost Function: -

#Input weights w1, w2, w3 as fractions/decimals. Usually w2 is largest.

$Cost(r) = w1 * D(r) + w2 * TWV(r) + w3 * CV(r)$

$w1 + w2 + w3 = 1$

D(r)=route duration

TWV(r)= number of time violations

CV(r)= number of capacity violations

Sequential Construction: -

Sort all requests wrt distance between depot, pick up point in descending order.

M=number of vehicles used. (initialize with 0)

Repeat (for all requests)

Initialize an empty route 'r'

M+=1

For all unassigned requests:

Get next unassigned request 'i'

Insert 'i' at the end of 'r'

Improve 'r' #use above heuristic

If r is feasible:

Insert 'i'

Else:

Remove 'i'

Order Keys (ID): -

Oid= order id (set 0=depot)

Pid = Pick up node Id

Did = Deliver node Id

Dist = Distance b/w pick up, depot

Rid =Route Id

x, y = locations

nid = node Id

demand = request size/quantity (+ (pick up), - (delivery))

twopen, twclose = Time window limits

Route: -

Id = route id

Path [] = ordered list of all nodes

Order [] = list of orders corresponding to the nodes

Update = flags for D, TWV, CV

appendorder(order)

heuristic function ()

update ()

cost function ()

Tabu Search: -

'Best of neighbourhood' can be used.

