**Main code:**

N = 81;

aa = 1;

aaa = 1;

JBAC = [];

JBACmax = [];

JBCAC = [];

AAACCC = [];

MMAAXX =[];

AAE =[];

MMEANAS = [];

NSNAS = [];

AASC = [];

ACS2 = [];

SSNAS = [];

AACCUmax = [];

lengthcollection = [];

RRAS = [];

MINVALUES = [];

ARGMINS = [];

originalAS = [];

for j = 1:N;

[c1,ia1,ic1] = unique(SS(:,27));%index the number code per location

bb = ia1(j+1)-1;

AS = SS(aa:bb,16);% actual start per location per day

AE = SS(aa:bb,17);% actual end per location per day

AC = AE - AS; % Actual change

meanAS = mean(AS);% Average actual start per location per day

MEANAS = round(meanAS);% round the average actual start to integer

j = j+1;% go to caculate the next location

aa = bb+1;

if bb >= 7133

break

end;

for k = -134:134;% The range of adding and deleting boxes to the orignial actual start

ASC = MEANAS + k; % MEANAS is the original AS, this is making it increase and decrease.

NAS = max(AS,ASC);% Select the bigger one between original actual start and after adding or deleting actual start.

SNAS = sum(NAS);% Total actual start per location

ACmax = AE - NAS;% Actual change per location per day

ACCmax = abs(ACmax);

ACRmax = (ACmax+ACCmax)/2;

ACCUmax = sum(ACRmax);% Sum of new positive AS

AACCUmax = [AACCUmax,ACCUmax];

ACC = abs(AC);

ACR = (AC+ACC)/2; % Positive actual change

ACCU = sum(ACR);% Sum of positive AS

ACS = sum(AS);% Sum of actual sart per location per day

AAEE = AE';

MAX = max(AE);

MMAAXX =[MMAAXX,MAX];

NSNAS = [NSNAS,SNAS];% inventory = sum of new changed AS

AASC = [AASC,ASC]; % collection of start AS increase and decrease

end;

originalAS = [originalAS,MEANAS];% Original average AS per location

JBAC = [JBAC,ACCU];%bobtail out per location

JBCAC = [JBCAC;ACS];% sum of actual start

end;

JBAM = [];

SSM(isnan(SSM(:,26)),:) = [];% delete the rows whose average mile is blank

for m = 1:N

[c11,ia11,ic11] = unique(SSM(:,27));% Index the codenumber

bbb = ia11(m+1)-1;

AM = mean(SSM(aaa:bbb,26));% Average the miles per location

m = m+1;

aaa = bbb+1;

if bbb >= 4114;

break

end;

AA = repmat(AM,269);

JBAM = [JBAM,AA];

RJBAM = JBAM(1,:);

end;

BOBTAILCOST = 5\*RJBAM.\*AACCUmax;% Bobtail cost per location, 2.5\*2\*average mile\*new positive actual changes

INVENTAORYCOST =40\*NSNAS;% inventory cost per location, inventory holding cost\*sum of new AS

TOTALCOST = BOBTAILCOST + INVENTAORYCOST;% Total cost = bobtailcost + inventory cost per location

TOTAL = [AASC;TOTALCOST];% Match the total new actual start with the total cost

R = find(AASC >= 0);% Only consider the situation that new changed AS is positive

final = TOTAL(:,R);% Find the total cost that new changed AS is positive

RR = find(final(1,:) == 0);% Find every start place with new changed AS=0

for m = 1:N-1

length = RR(m+1) - RR(m);% Find the length of new changed AS in every location

lengthcollection = [lengthcollection,length];

end;

for t = 1:N-1

q = RR(t);% The start point of new changed AS for location t

qq = RR(t)+lengthcollection(t)-1;% The end point of new changed AS for location t

RAS = [final(:,q:qq)];% divide all the new changed AS and all the cost along t locations

RRAASS = RAS(2,:);% The arry of total cost

[minValues, argmins] = find\_minimum(RRAASS);% Call the find\_minimum function we defined to find all the local minimal total cost

argmins = argmins - 1;

if size(argmins, 2) > 1% If there is more than 1 local minimal total cost

u = size(argmins, 2);% The number of local minimal total cost

w = originalAS(t);% Original average AS of location t

y = repmat(w,1,u);

e = abs(argmins-y);%Find the distance from minimal local cost to original cost

argmins = argmins(min(e) == e);% Find the minimal distance from minimal local cost to original cost

idx = min(RRAASS(argmins))== RRAASS(argmins);% Find the minimal total cost(there may be several points that have same minimal distance from minimal local cost to original cost)

argmins = argmins(idx);% Return the optimal changed AS

minValues = RRAASS(argmins);% Return the optimal minimal total cost

end

ARGMINS = [ARGMINS,argmins];% Save all values of optimal changed AS of t locations

MINVALUES = [MINVALUES,minValues];% Save all values of optimal minimal total cost of t locations

end

MINTOTALCOST = reshape(TOTALCOST,269,81);

originalAS = originalAS(1:80);

originalAS = originalAS';

MM = ARGMINS'-originalAS;

rampgroupbanlance = sum(MM(:,1))% Add all changing boxes up of t locations