function plotOPbyuser(LastChangeVector, divider)

%LastChangeVector= Last Change Vector

%divider= # of n skips when graphing

if (nargin < 2)

divider = ceil(length(LastChangeVector)/500);

%default divider is the ceiling of N/500

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Making Underlying Optimal Partition Graph%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for x=1:divider:length(LastChangeVector)

%x=n

y= LastChangeVector(x);

%y=first changepoint of n=x

while y>1

plot(x,y, '.')

y = LastChangeVector(y-1);

%plots each individual optimal changepoint for x=n

end

plot(x,1, '.')

%last optimal changepoint is 1

hold on

end

grid minor

xlabel('n')

ylabel('Optimal Changepoints')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%end of underlying plot%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Marking 1st ecounter of new Optimal changepoint with red%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

firstcplist=1;

%firstcplist=array that starts the unique changepoints encountered

j=1;

%initializes the j variable

for x=1:divider:length(LastChangeVector)

n\_OP=LastChangeVector(x);

while j>1

j=LastChangeVector(LastChangeVector(j)-1);

n\_OP=[j n\_OP]; %#ok<AGROW> %stores optimal partition of n data points

end

for j=1:length(n\_OP) %find if element j has been expressed already in previous optimal partitions of n-1 pieces of data

check=0;

for k=1:length(firstcplist)

if n\_OP(j)==firstcplist(k) %loop compares element j with each element in firstcplist to find repititions

check=1;

end

end

if check==0

firstcplist=[firstcplist n\_OP(j)]; %#ok<AGROW> %stores the unique lastchangepoint

plot(x,n\_OP(j),'r\*') %plots the unique lastchangepoint on the graph, if any

end

end

end

plot(0,0,'r\*')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%end of Marking%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hold off