```
function [final_label_vector delta] = label_based_optimize_clusters_BlackTree(dataset,
N)

num_rows = size(dataset,1);
is_available = ones(num_rows,1);
final_label_vector = -1*ones(num_rows,1);

%initial iteration occurs outside the main loop3prediction

s = std(dataset(:,1:N)); %calculates the standard deviation of the dataset in each
dimension
s = mean(s); %takes the average standard deviation
alpha = 1.5; %this is a constant used to adjust the standard deviation
s = s*alpha;

num_iterations = 25;
increment = s/num_iterations;
delta_vector = [increment : increment : s];
delta = delta_vector(1);

%generates the labels for each row
[prior_label_vector] = label_based_generate_clusters_BlackTree(dataset, delta, N);

%finds the entropy of the label vector
[prior_frequency_vector] = get_label_frequencies_BlackTree(prior_label_vector,
num_rows);

i = 1;
max_change = 0;
break_loop = 0;

while(i <= num_iterations && !break_loop)

  delta = delta_vector(i);
  [current_label_vector] = label_based_generate_clusters_BlackTree(dataset, delta, N);
  current_frequency_vector = get_label_frequencies_BlackTree(current_label_vector,
num_rows);

  change_in_structure = sum(abs(prior_frequency_vector .- current_frequency_vector));

  %if true, then we store the related values
  if(change_in_structure > max_change)

    max_change = change_in_structure;
    final_label_vector = current_label_vector;
    final_delta = delta_vector(i);
  endif

  i = i + 1;
  prior_label_vector = current_label_vector;
  prior_frequency_vector = current_frequency_vector;

  a = max(prior_frequency_vector);

  %if true, then all rows are in one cluster, terminate
  if(a == num_rows)

    break_loop = 1;

  endif

endwhile

endfunction
```