```
%copyright Charles Davi 2022

function weight_matrix = Mem_Optimization_Fill_Matrix_GB(problem_domain,
search_depth)

  num_rows = size(problem_domain,2);

  num_dimensions = size(problem_domain,1);

  weight_matrix = ones(num_rows,search_depth,num_dimensions)*Inf;
  x = find(weight_matrix(:) == Inf);
  num_unknown = size(x);

  %iterates until the weight_matrix is filled
  while(num_unknown > 0)

    %iterates over all dimensions
    for D = 1 : num_dimensions

      current_pos(D) = randi(num_rows);

    endfor %end of D-loop

    %evaluates polynomial and tests difference from goal
    [approx_goal approx_quantity] = eval_goal_function(problem_domain, current_pos);

    %updates the weights for each dimension
    for D = 1 : num_dimensions

      %populates the first available column with the weight
      x = find(weight_matrix(current_pos(D),:,D) == Inf);

      %if empty we ignore
      if(!isempty(x))

        weight_matrix(current_pos(D),x(1),D) = approx_goal;

      endif

    endfor

    x = find(weight_matrix(:) == Inf);
    num_unknown = size(x,1);

  endwhile %end of outer k-loop

endfunction
```