```
    lambda = c/nuHZ;



    % A matrix is used to ensure that the maximum_elements is not exceeded
    A = [];

        % For loop to make a linear constraint for the number of antennas.
        % The sum of all of the antennas must be less than or equal to the
        % maximum_elements (b vector).
        for j = 1:num_styles
            A = [A,0,1,0,0];
        end

    % bounding vector
    b = [max_antennas];

    % equality contraints (not presently used)
    Aeq = [];
    beq = [];


    % calls bounding function to set lower and upper bounds of each parameter
    [lb,ub] = bounding(num_parameters,num_styles,min_diameter,max_diameter,min_power, ↙
max_power,min_quantity,max_quantity,min_t_styles,min_r_styles);

    %↙
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ↙
~~~~~~~~~~~~~~~~
    % gamultiobj is a built in function in Matlab that uses a genetic algorithm
    % to solve multi-variable minimization optimization problems with various
    % constraints. This program uses gamultiobj to find a pareto curve of
    % solutions for radar array design.
    % for more information about the gamultiobj function see:
    % https://www.mathworks.com/help/gads/gamultiobj-algorithm.html#mw_c77c838e-b703- ↙
437f-9d91-a26d5927d65b
    %↙
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ↙
~~~~~~~~~~~~~~~~



    % calls the gamultiobj function
    [x,fval] = gamultiobj(@objectives, num_vars, A, b, Aeq, beq, lb, ub, options);


        % nested function that calls the objective functions to calculate cost
        % and gain.
        % needed by gamultiobj

        function f = objectives(x)
            [f(1),f(2)] = get_objectives_function(x,year_built,num_styles,k,lambda); ↙
% cost and gain
        end % end function f = ojectives(x)
```