

```

% table = table of optimized solution
% desired_loop_gain: input from user in main_life_script to determine
% the upper and lower horizontal lines.
% ~~~~~
% Return
% displayed scatterplot graph
% ~~~~~

function create_graph(dispatch_table,loop_gain_desired)

    % variables to hold the upper and lower line values
    upper_line = loop_gain_desired*1.1;
    lower_line = loop_gain_desired*.9;

    % Create scatter of table.Cost and table.Gain
    scatter_graph = scatter(dispatch_table, "Cost ($M)", "Gain (dB,dBW)");

    % Add xlabel, ylabel, title, legend and lower and upper gain lines
    xlabel("Cost ($M)")
    ylabel("Gain (dB,dBW)")
    title("Gain vs. Cost")
    yline(upper_line, '-.g')
    yline(lower_line, '-.g')

    % Customizing data tips
    dcm = datacursormode(gcf);
    set(dcm, 'UpdateFcn', @(src, event_obj) customDataTip(event_obj, \
dispatch_table));
    datacursormode on; % Turn on the data cursor mode

    % Inner function for custom data tip text
    function txt = customDataTip(event_obj, dispatch_table)
        pos = event_obj.Position;

        % Find the index (row number) of the closest point
        distances = hypot(dispatch_table.("Cost ($M)") - pos(1), dispatch_table.("Gain \
(dB,dBW)") - pos(2));
        [~, row] = min(distances);

        % Create the data tip text
        txt = {['Solution: ', num2str(row)], ...
                ['Cost ($M)', num2str(dispatch_table.("Cost ($M)")(row))], ...
                ['Gain: ', num2str(dispatch_table.("Gain (dB,dBW)")(row)), ' dB, \
dBW']};
    end

end % t_output_graph function

function saveFile(dispatch_table)

    list_parameter_names = ["Number of Styles"; "Minimum Transmitter Styles"; \
"Minimum Recievers Styles"; "Minimum Quantity"; "Maximum Quantity"; "Minimum Diameter"; \
Max Diameter"; "Minimum Power" ; "Maximum Power"; "Year to be Built"; "Include \

```