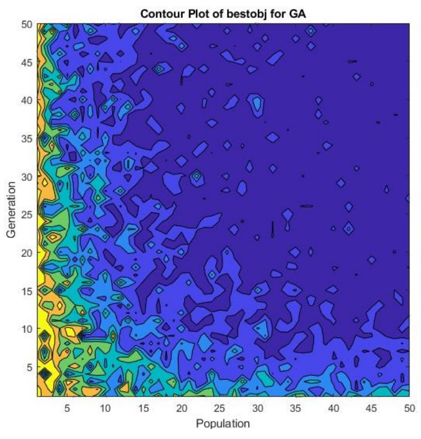
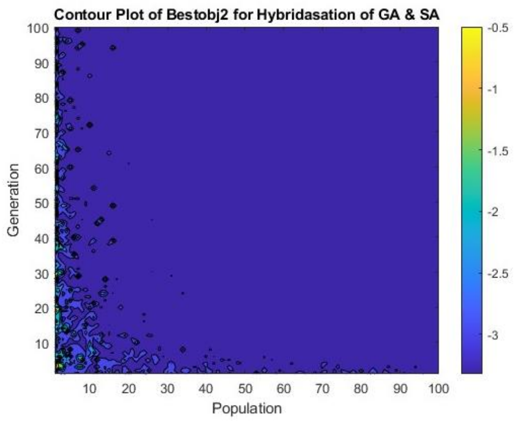
Question 1

Accuracy & Consistency Investigation



(a)



(b)

Figure 1: Contour plot of best objective residual using (a) Genetic Algorithm (GA) constructed using Test_function3. (b) Hybridization of GA and Fixed 20 SA constructed using Test_function3.

Figure 1 (a)(b) demonstrated the trade-off between population and generation count to achieve an optimal minimal objective value. As the lighter region of the contour focus more alongside the generation axis than the population axis, it is suggested that with higher generation and lower population, the objective value is less optimal. This finding stays true for only having GA and also hybridized optimizer. Hence stressing for higher importance in population over generation towards the accuracy of the optimizer.

To investigate the consistency, the below table 1 is constructed by performing 10 runs on each combination of population and generation.

Population	Generation	Standard Deviation	Mean Value
		GA	GA
20	5	0.34073976	-2.33411
5	20	0.76828462	-1.69984

Table 1: Comparison between different ratios of population to generation.

It is evidence that the higher population case gives a lower standard deviation and mean value. Thus, a higher population to generation for the GA scheme should be implement for higher accuracy and consistency.

Hybridized Optimizer Design

Figure 2 below is constructed to investigate the ratio between the GA and Simplec Algorithm (SA) for the hybridized optimizer for an optimal achievable value of the best objective function. Notably, the total runs of GA on the x-axis also reflects the total runs of SA as they are related by *Eqn. 1* below. The best objective is obtained using the mean of best objectives from all combinations between population and generation that form the particular total runs of GA.

$$Total\ Runs\ of\ SA\ =\ 120\ -\ Total\ Runs\ of\ GA \qquad \text{Eqn. 1}$$

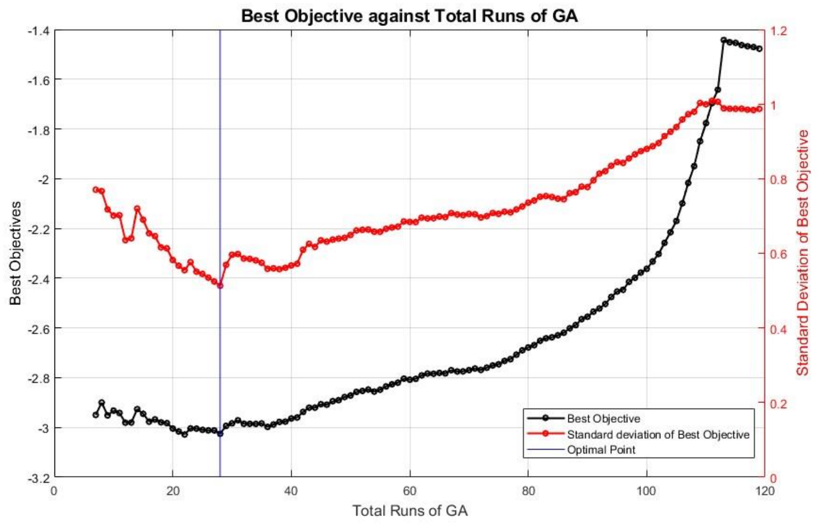


Figure 2: Best Objective against Total Runs of GA in a Total of 120 Runs of GA and SA.

It is evident that the within the range of 20 GA, 100 SA and 40 GA, 80SA, both the final best objectives and the standard deviations of it are plausible. The absolute best solution within the range is marked across the blue line, with parameter as shown below table 2.

Global Runs	Local Runs	Best Objectives	Standard Deviation
28.0000	92.0000	-3.0252	0.5132

Table 2: Details of the Optimal Point from Figure 2.

Further investigating into the combination of population and generation to achieve 28 total GA runs, the following figure 3 is constructed. Each combination is iterated for 100 times using the hybridized optimizer for the mean and standard deviation to be calculated. Notably, the population size and generation count in this context is related by *Eqn. 2* below.

$$Generation\ Count\ =\ \frac{Total\ GA\ Runs}{Population\ Size} \qquad \text{Eqn. 2}$$

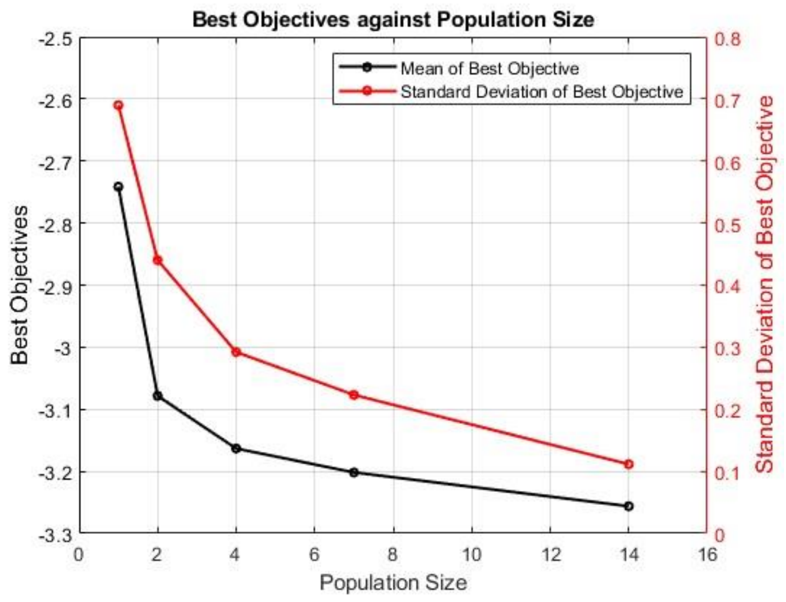


Figure 3: Best Objective against Population Size.

Hence, the optimal combination of the population and generation should be 14 and 2 respectively. The conclusion also agrees with finding in section above, stating priority of population over generation for accuracy and consistency.

Wing Design Optimizer 1

The settings for the optimizer for Q1 are as presented in table 3.

Genetic Algorithm		Simplec Algorithm
Population	Generation	
14	1 + 1 = 2	94

Table 3: Settings for Optimizer.

The designed output from the optimizer is then found to be as table 4.

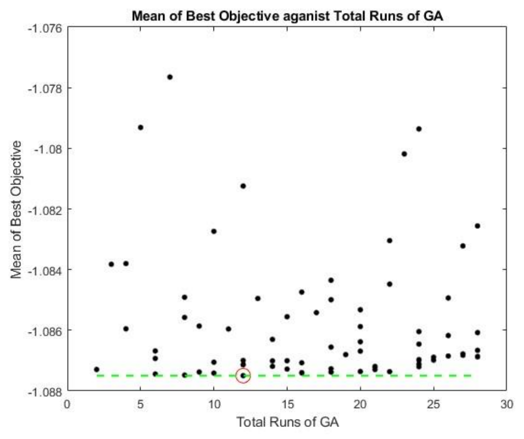
Best Objective	Best Variables						
	Aspect Ratio	Area	Taper Ratio	Airfoil Shape			
0.0117	0.9970	0.9999	0.4273	0.7723	0.0433	0.2424	0.3462
Denormalized:	11.9819	19.9994	0.5419	0.0054	-		

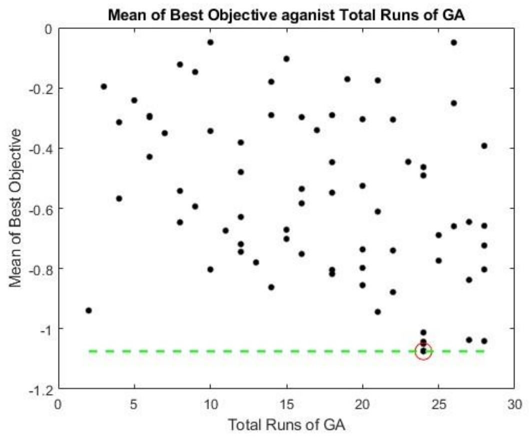
Table 4: Output Results from the Optimizer.

Question 2

Separate GA Parameter Investigation

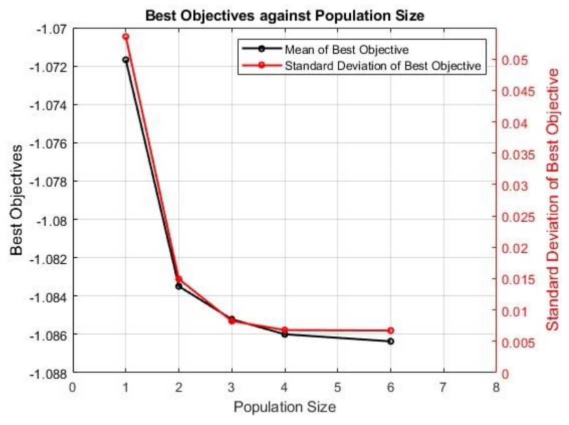
Figure 4 is constructed to investigate the total runs of GA needed when the variables that requires optimization is reduced. The maximum total GA runs are cap at 28 as it would be unsensible to require more runs than the case of optimizing all 6 variables in question 1.

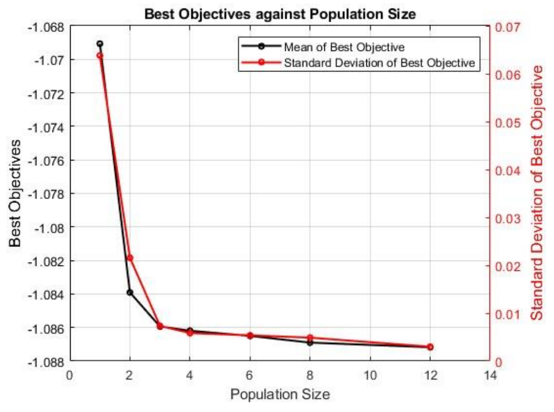




(a) (b)

Figure 4: The Mean of Best Objectives against Total Runs of GA for optimizing (a) 3 out of 6 variables only. (b) 4 out of 6 Variables only.





(a) (b)

Figure 5: Best Objectives against Population Size for Total GA count of (a) 12 (b) 24.

Figure 4(a) depicted that for optimization of 3 variables, a total of 12 GA runs will be sufficient. Figure 4(b) suggested that a total of 24 runs for optimization of 4 variables.

Further investigation regarding the combination of population to generation for optimal accuracy and stability of the 3 and 4 variables case are conducted. Figure 5(a)(b) are produced using similar method toward figure 3. As it demonstrated, the best combination to achieve 12 and 24 GA runs will be “6 Populations, 2 Generations” and “12 Populations, 2 Generations” respectively.

Ratio of Allocated Runs

Figure 6 is constructed to investigate the best allocation ratios to split the optimizer. In this context, “Test Function 3” are split into optimizing 2 variables on the first half, followed by the other 4 on the second half. The settings of first and second half GA are as investigated on above section. Hence the relation between the first and second half’s SA runs are as *Eqn.3* below.

$$Second\ Half\ Runs\ of\ SA = 120 - 12 - 24 - First\ Half\ Runs\ of\ SA \qquad \text{Eqn. 3}$$

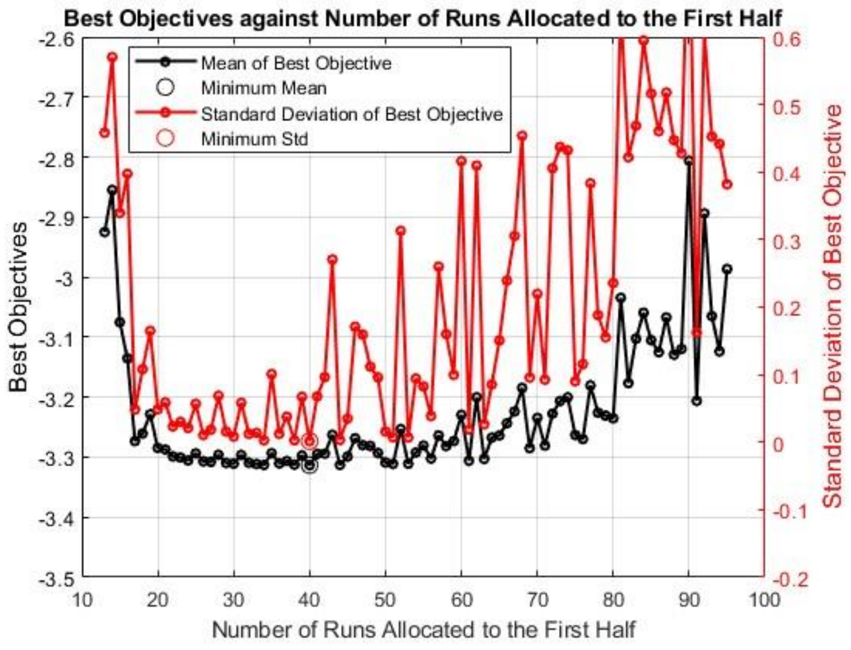


Figure 6: Best Objectives against Number of Runs among 120 Allocated to the First Half.

Table 6 presents the details of the optimal points regarding their accuracy and consistency based on figure 6 above.

First Half's Runs	Second Half's Runs	Best Objective	Standard Deviation
40	80	-3.31325	0.00156247

Table 5: Details of the optimal points from figure 4.

Wing Design Optimizer 2

Hence, the settings for the new optimizer for question 2 are as table 7.

Optimizer for First Half			Optimizer for Second Half		
Genetic Algorithm		Simplec Algorithm	Genetic Algorithm		Simplec Algorithm
Population	Generation		Population	Generation	
6	1 + 1 = 2	28	12	1 + 1 = 2	56

Table 6: Settings for Optimizer.

The design output obtained from the new optimizer is as displayed in table 8.

Best Objective	Best Variables						
	Aspect Ratio	Area	Taper Ratio	Airfoil Shape			
0.0115	0.9975	0.9985	5.739E-4	0.0001	0.2648	0.0746	0.1196

Table 7: Output Results from the Optimizer.

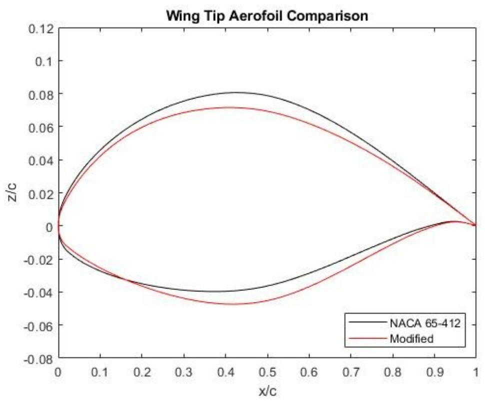
Question 3

		Baseline	Optimizer 1	Optimizer 2
Parameters	**Aspect Ratio**	7.5	12	12
	Wing Area (m^2)	16	20	20
	Taper Ratio	0.4	0.542	0.2
Performance	**Lift Coefficient**	0.342	0.290	0.288
	Total Drag Coefficient	0.0147	0.0117	0.0115
	Vortex Drag Coefficient	0.0056	0.0024	0.0024
	Viscous Drag Coefficient	0.0091	0.0093	0.0091
	Lift to Drag Ratio	23.3	24.8	25

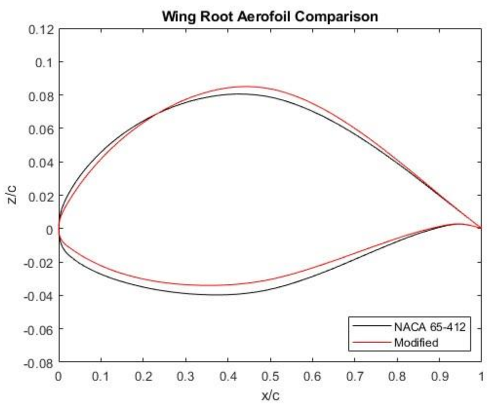
Table 8: Parameter of the Baseline, First Optimizer's, and Second Optimizer's Designs.

The goal to the optimizer is set such that the drag produced are minimized. Hence despite the reduced lift coefficient in both optimizers, the total drag coefficient is reduced in both cases as shown in table 9. Even with a lower lift coefficient, the reduction in total drag coefficient has result in a better lift to drag ratio thus efficiency. This depicted the successful process of the optimizer to allocate the limited resources correctly and returning an optimized design.

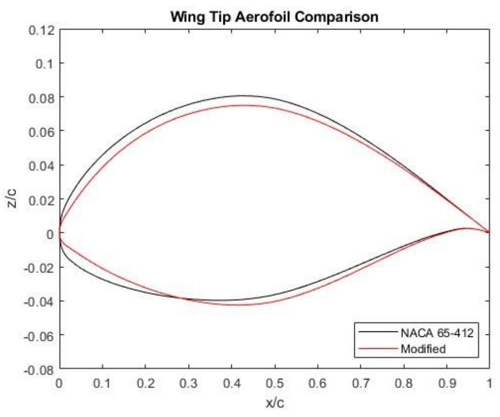
Notably, the second optimizer has a slightly better performance over the first optimizer. This is evident by the trade-off of minor reduction in lift coefficient for a smaller viscous drag coefficient, resulting smaller total drag coefficient.



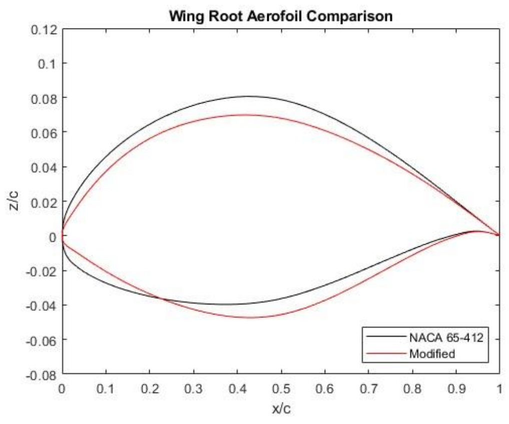
(a)



(b)



(c)



(d)

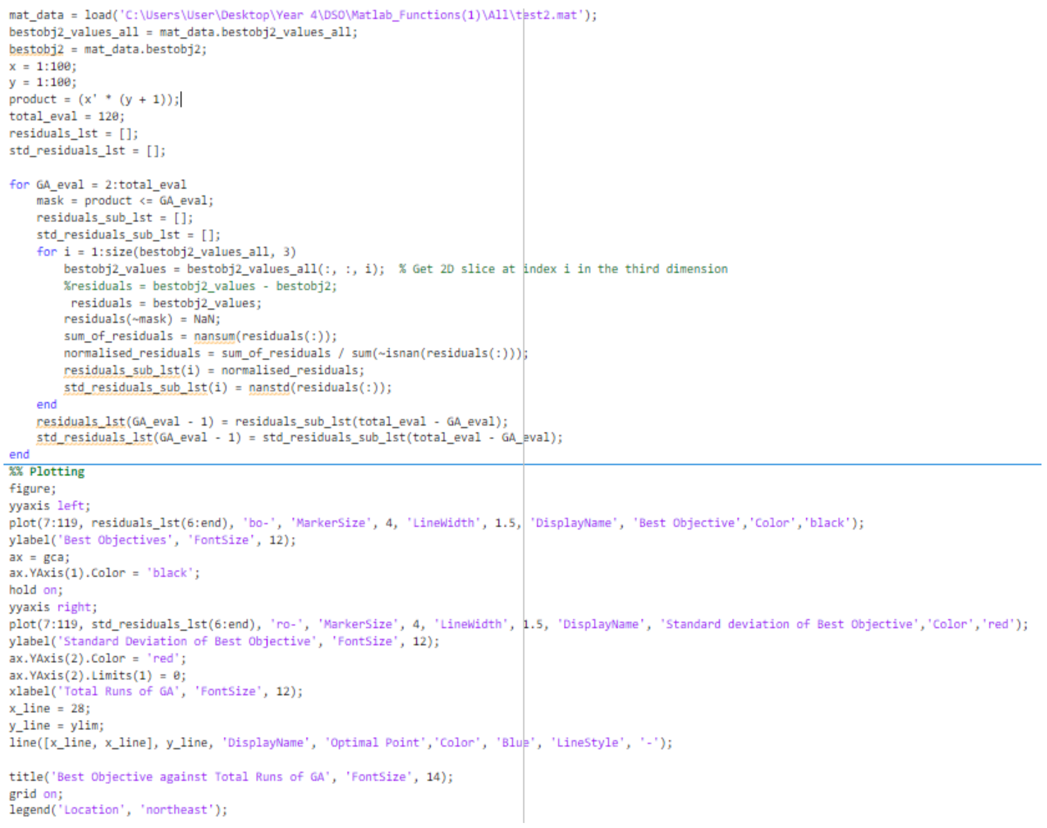
Figure 7: First Optimized Design's (a) Wing Tip (b) Wing Root Aerofoil Shape.

Second Optimized Design's (c) Wing Tip (d) Wing Root Aerofoil Shape.

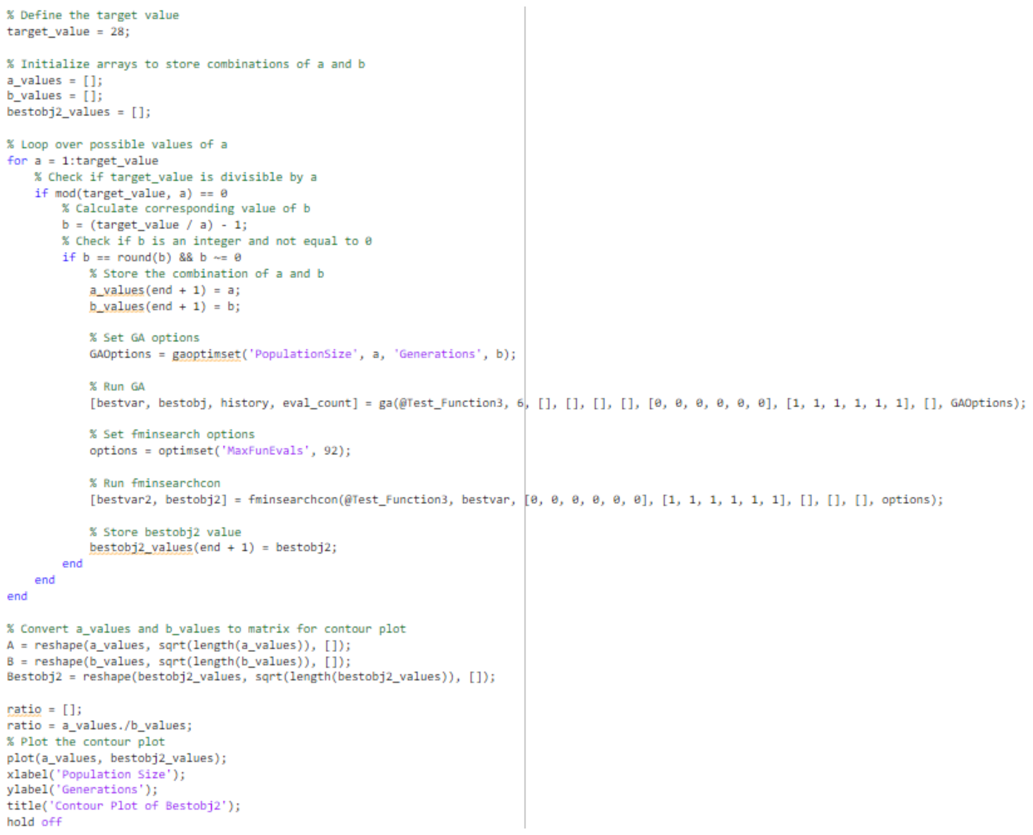
As depicted from figure 7, both optimizers altered the shape of the aerofoil which results in a visible change when compared back to the baseline wing. It is also obvious that the first and second optimizer provides very different wing root aerofoil. Combined with the setting angle of the wing, these might be the reason behind the improvement of drag.

Appendix

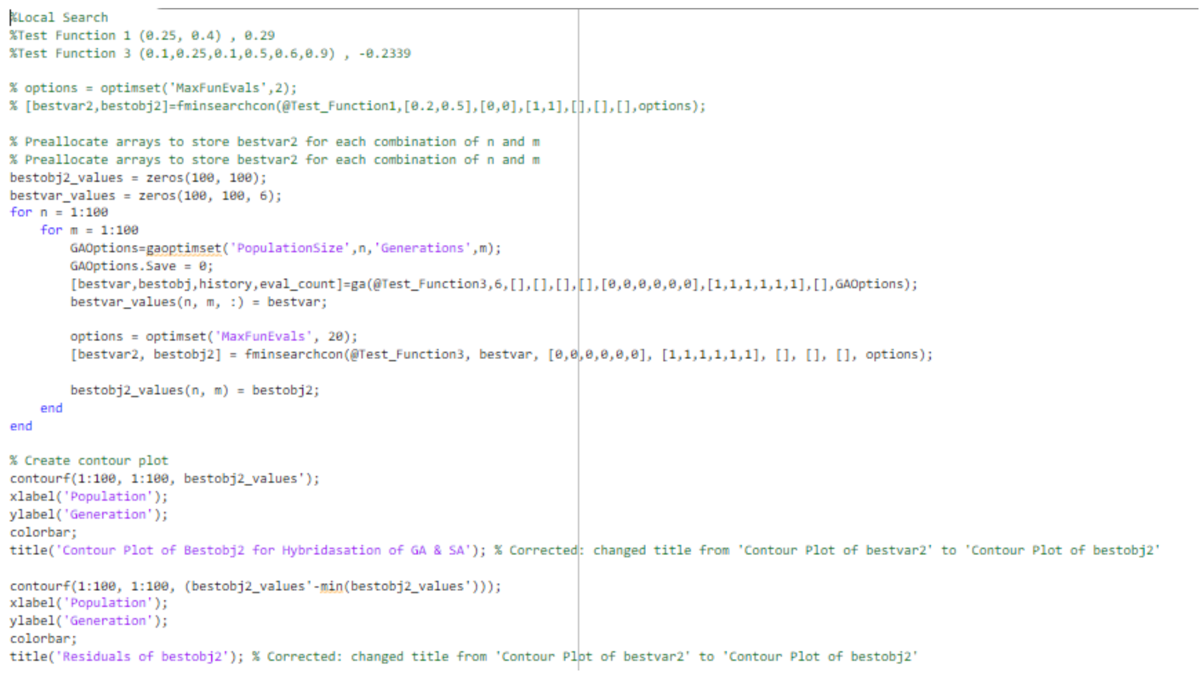
Question 1



```matlab
mat_data = load('C:\Users\User\Desktop\Year 4\DSO\Matlab_Functions(1)\All\test2.mat');
bestobj2_values_all = mat_data.bestobj2_values_all;
bestobj2 = mat_data.bestobj2;
x = 1:100;
y = 1:100;
product = (x' * (y + 1));
total_eval = 120;
residuals_lst = [];
std_residuals_lst = [];

for GA_eval = 2:total_eval
    mask = product <= GA_eval;
    residuals_sub_lst = [];
    std_residuals_sub_lst = [];
    for i = 1:size(bestobj2_values_all, 3)
        bestobj2_values = bestobj2_values_all(:, :, i);  % Get 2D slice at index i in the third dimension
        %residuals = bestobj2_values - bestobj2;
         residuals = bestobj2_values;
        residuals(~mask) = NaN;
        sum_of_residuals = nansum(residuals(:));
        normalised_residuals = sum_of_residuals / sum(~isnan(residuals(:)));
        residuals_sub_lst(i) = normalised_residuals;
        std_residuals_sub_lst(i) = nanstd(residuals(:));
    end
    residuals_lst(GA_eval - 1) = residuals_sub_lst(total_eval - GA_eval);
    std_residuals_lst(GA_eval - 1) = std_residuals_sub_lst(total_eval - GA_eval);
end
%% Plotting
figure;
yyaxis left;
plot(7:119, residuals_lst(6:end), 'bo-', 'MarkerSize', 4, 'LineWidth', 1.5, 'DisplayName', 'Best Objective','Color','black');
ylabel('Best Objectives', 'FontSize', 12);
ax = gca;
ax.YAxis(1).Color = 'black';
hold on;
yyaxis right;
plot(7:119, std_residuals_lst(6:end), 'ro-', 'MarkerSize', 4, 'LineWidth', 1.5, 'DisplayName', 'Standard deviation of Best Objective','Color','red');
ylabel('Standard Deviation of Best Objective', 'FontSize', 12);
ax.YAxis(2).Color = 'red';
ax.YAxis(2).Limits(1) = 0;
xlabel('Total Runs of GA', 'FontSize', 12);
x_line = 28;
y_line = ylim;
line([x_line, x_line], y_line, 'DisplayName', 'Optimal Point','Color', 'Blue', 'LineStyle', '-');

title('Best Objective against Total Runs of GA', 'FontSize', 14);
grid on;
legend('Location', 'northeast');
```

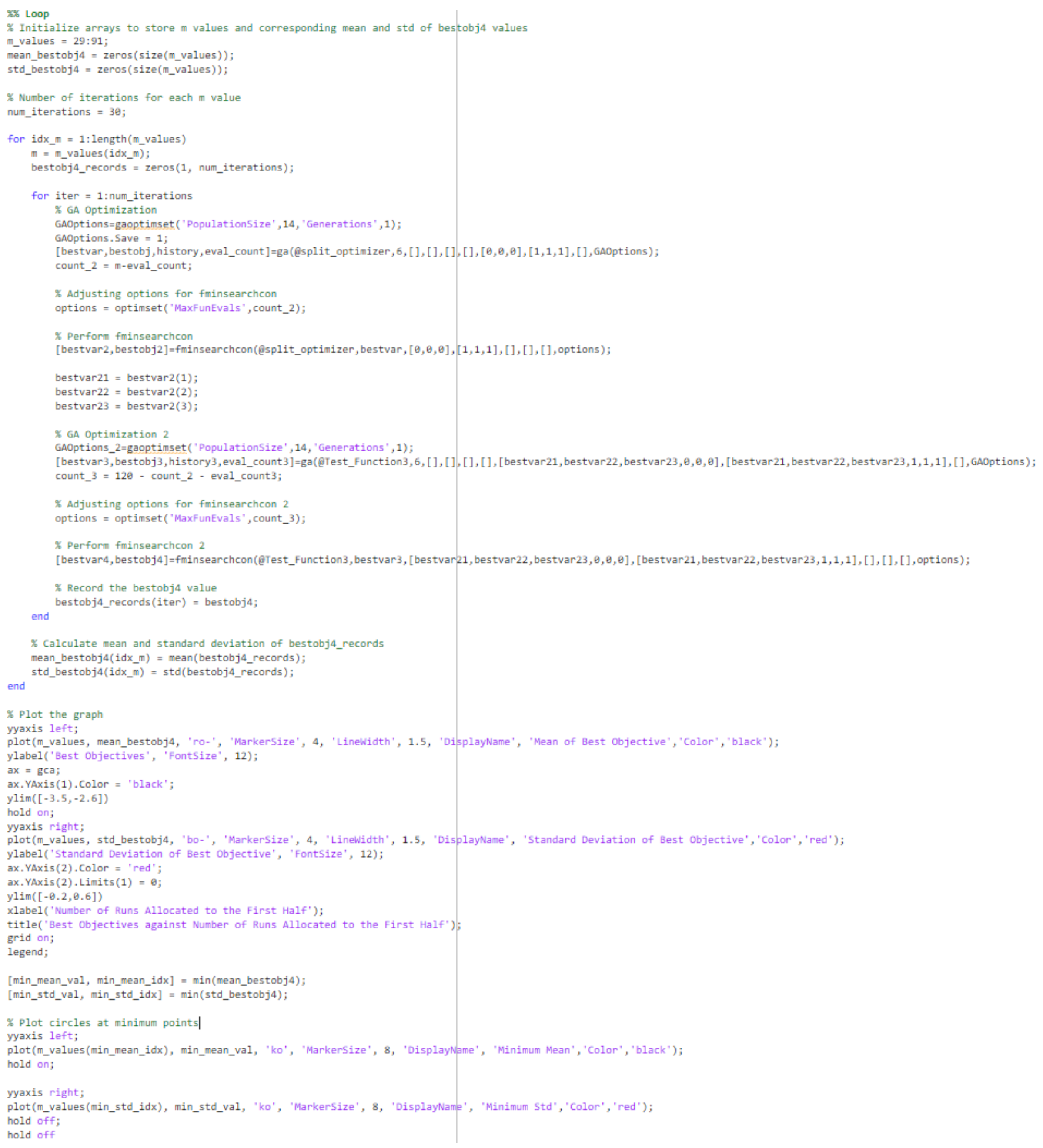


```matlab
% Define the target value
target_value = 28;

% Initialize arrays to store combinations of a and b
a_values = [];
b_values = [];
bestobj2_values = [];

% Loop over possible values of a
for a = 1:target_value
    % Check if target_value is divisible by a
    if mod(target_value, a) == 0
        % Calculate corresponding value of b
        b = (target_value / a) - 1;
        % Check if b is an integer and not equal to 0
        if b == round(b) && b ~= 0
            % Store the combination of a and b
            a_values(end + 1) = a;
            b_values(end + 1) = b;

            % Set GA options
            GAOptions = gaoptimset('PopulationSize', a, 'Generations', b);

            % Run GA
            [bestvar, bestobj, history, eval_count] = ga(@Test_Function3, 6, [], [], [], [], [0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 1], [], GAOptions);

            % Set fminsearch options
            options = optimset('MaxFunEvals', 92);

            % Run fminsearchcon
            [bestvar2, bestobj2] = fminsearchcon(@Test_Function3, bestvar, [0, 0, 0, 0, 0, 0], [1, 1, 1, 1, 1, 1], [], [], [], options);

            % Store bestobj2 value
            bestobj2_values(end + 1) = bestobj2;
        end
    end
end

% Convert a_values and b_values to matrix for contour plot
A = reshape(a_values, sqrt(length(a_values)), []);
B = reshape(b_values, sqrt(length(b_values)), []);
Bestobj2 = reshape(bestobj2_values, sqrt(length(bestobj2_values)), []);

ratio = [];
ratio = a_values./b_values;
% Plot the contour plot
plot(a_values, bestobj2_values);
xlabel('Population Size');
ylabel('Generations');
title('Contour Plot of Bestobj2');
hold off
```

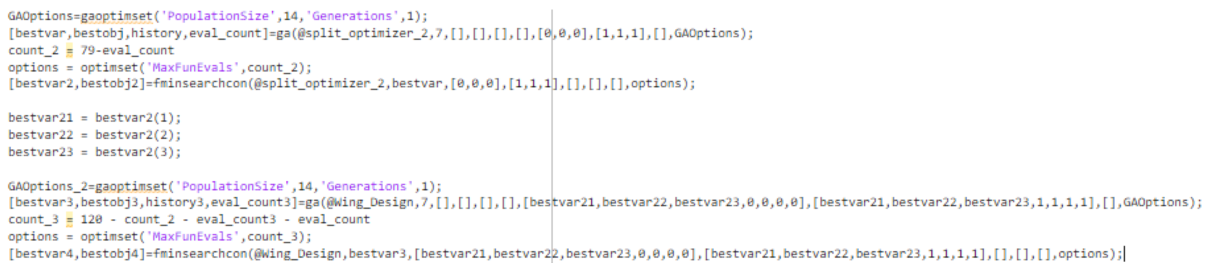


```
%Local Search
%Test Function 1 (0.25, 0.4) , 0.29
%Test Function 3 (0.1,0.25,0.1,0.5,0.6,0.9) , -0.2339

% options = optimset('MaxFunEvals',2);
% [bestvar2,bestobj2]=fminsearchcon(@Test_Function1,[0.2,0.5],[0,0],[1,1],[],[],[],options);

% Preallocate arrays to store bestvar2 for each combination of n and m
% Preallocate arrays to store bestvar2 for each combination of n and m
bestobj2_values = zeros(100, 100);
bestvar_values = zeros(100, 100, 6);
for n = 1:100
    for m = 1:100
        GAOptions=gaoptimset('PopulationSize',n,'Generations',m);
        GAOptions.Save = 0;
        [bestvar,bestobj,history,eval_count]=ga(@Test_Function3,6,[],[],[],[],[0,0,0,0,0,0],[1,1,1,1,1,1],[],GAOptions);
        bestvar_values(n, m, :) = bestvar;

        options = optimset('MaxFunEvals', 20);
        [bestvar2, bestobj2] = fminsearchcon(@Test_Function3, bestvar, [0,0,0,0,0,0], [1,1,1,1,1,1], [], [], [], options);

        bestobj2_values(n, m) = bestobj2;
    end
end

% Create contour plot
contourf(1:100, 1:100, bestobj2_values');
xlabel('Population');
ylabel('Generation');
colorbar;
title('Contour Plot of Bestobj2 for Hybridasation of GA & SA'); % Corrected: changed title from 'Contour Plot of bestvar2' to 'Contour Plot of bestobj2'

contourf(1:100, 1:100, (bestobj2_values'-min(bestobj2_values')));
xlabel('Population');
ylabel('Generation');
colorbar;
title('Residuals of bestobj2'); % Corrected: changed title from 'Contour Plot of bestvar2' to 'Contour Plot of bestobj2'
```

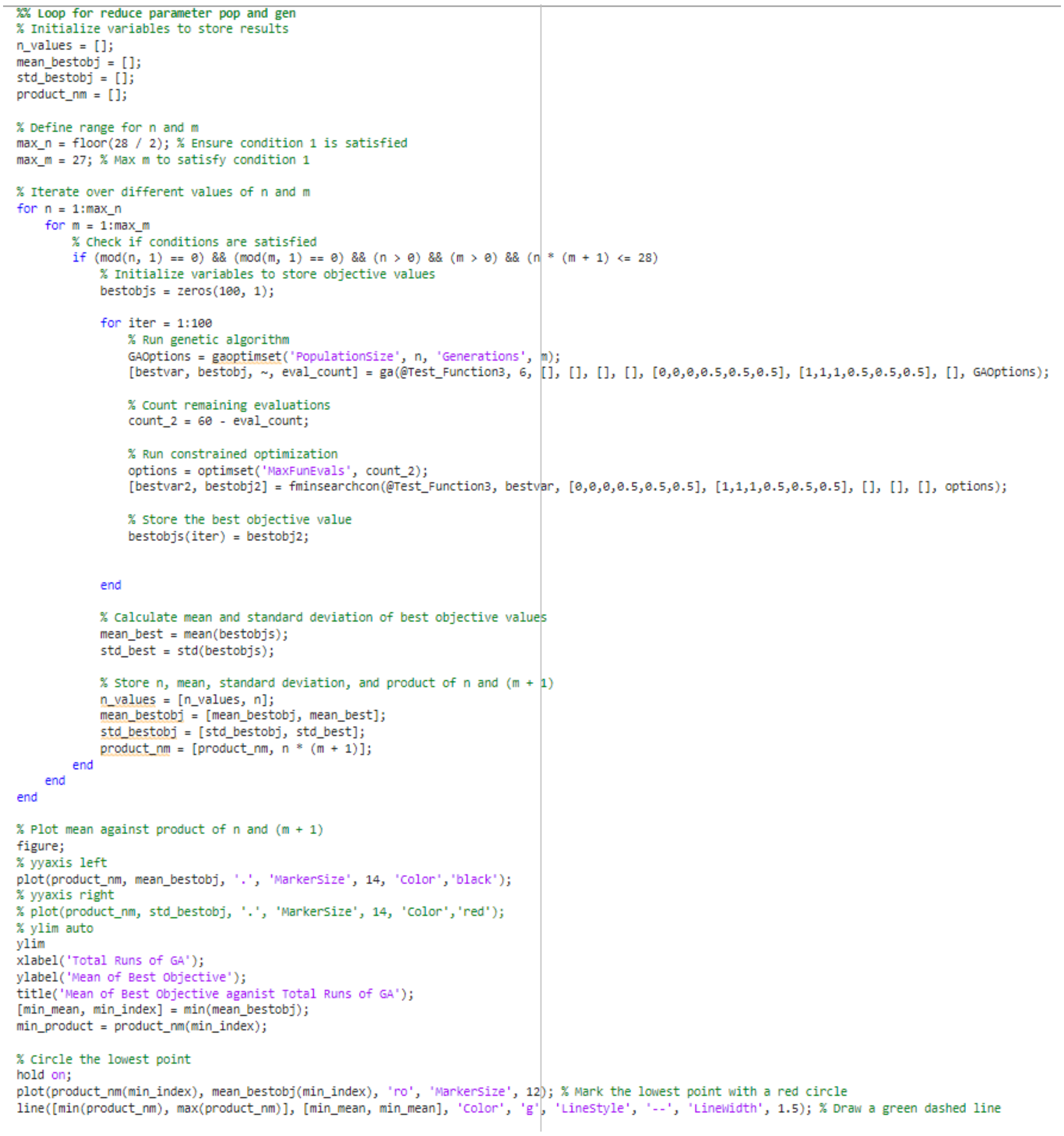
Question 2



```matlab
%% Loop
% Initialize arrays to store m values and corresponding mean and std of bestobj4 values
m_values = 29:91;
mean_bestobj4 = zeros(size(m_values));
std_bestobj4 = zeros(size(m_values));

% Number of iterations for each m value
num_iterations = 30;

for idx_m = 1:length(m_values)
    m = m_values(idx_m);
    bestobj4_records = zeros(1, num_iterations);

    for iter = 1:num_iterations
        % GA Optimization
        GAOptions=gaoptimset('PopulationSize',14,'Generations',1);
        GAOptions.Save = 1;
        [bestvar,bestobj,history,eval_count]=ga(@split_optimizer,6,[],[],[],[],[0,0,0],[1,1,1],[],GAOptions);
        count_2 = m-eval_count;

        % Adjusting options for fminsearchcon
        options = optimset('MaxFunEvals',count_2);

        % Perform fminsearchcon
        [bestvar2,bestobj2]=fminsearchcon(@split_optimizer,bestvar,[0,0,0],[1,1,1],[],[],[],options);

        bestvar21 = bestvar2(1);
        bestvar22 = bestvar2(2);
        bestvar23 = bestvar2(3);

        % GA Optimization 2
        GAOptions_2=gaoptimset('PopulationSize',14,'Generations',1);
        [bestvar3,bestobj3,history3,eval_count3]=ga(@Test_Function3,6,[],[],[],[],[bestvar21,bestvar22,bestvar23,0,0,0],[bestvar21,bestvar22,bestvar23,1,1,1],[],GAOptions);
        count_3 = 120 - count_2 - eval_count3;

        % Adjusting options for fminsearchcon 2
        options = optimset('MaxFunEvals',count_3);

        % Perform fminsearchcon 2
        [bestvar4,bestobj4]=fminsearchcon(@Test_Function3,bestvar3,[bestvar21,bestvar22,bestvar23,0,0,0],[bestvar21,bestvar22,bestvar23,1,1,1],[],[],[],options);

        % Record the bestobj4 value
        bestobj4_records(iter) = bestobj4;
    end

    % Calculate mean and standard deviation of bestobj4_records
    mean_bestobj4(idx_m) = mean(bestobj4_records);
    std_bestobj4(idx_m) = std(bestobj4_records);
end

% Plot the graph
yyaxis left;
plot(m_values, mean_bestobj4, 'ro-', 'MarkerSize', 4, 'LineWidth', 1.5, 'DisplayName', 'Mean of Best Objective','Color','black');
ylabel('Best Objectives', 'FontSize', 12);
ax = gca;
ax.YAxis(1).Color = 'black';
ylim([-3.5,-2.6])
hold on;
yyaxis right;
plot(m_values, std_bestobj4, 'bo-', 'MarkerSize', 4, 'LineWidth', 1.5, 'DisplayName', 'Standard Deviation of Best Objective','Color','red');
ylabel('Standard Deviation of Best Objective', 'FontSize', 12);
ax.YAxis(2).Color = 'red';
ax.YAxis(2).Limits(1) = 0;
ylim([-0.2,0.6])
xlabel('Number of Runs Allocated to the First Half');
title('Best Objectives against Number of Runs Allocated to the First Half');
grid on;
legend;

[min_mean_val, min_mean_idx] = min(mean_bestobj4);
[min_std_val, min_std_idx] = min(std_bestobj4);

% Plot circles at minimum points
yyaxis left;
plot(m_values(min_mean_idx), min_mean_val, 'ko', 'MarkerSize', 8, 'DisplayName', 'Minimum Mean','Color','black');
hold on;

yyaxis right;
plot(m_values(min_std_idx), min_std_val, 'ko', 'MarkerSize', 8, 'DisplayName', 'Minimum Std','Color','red');
hold off;
hold off
```



```matlab
GAOptions=gaoptimset('PopulationSize',14,'Generations',1);
[bestvar,bestobj,history,eval_count]=ga(@split_optimizer_2,7,[],[],[],[],[0,0,0],[1,1,1],[],GAOptions);
count_2 = 79-eval_count
options = optimset('MaxFunEvals',count_2);
[bestvar2,bestobj2]=fminsearchcon(@split_optimizer_2,bestvar,[0,0,0],[1,1,1],[],[],[],options);

bestvar21 = bestvar2(1);
bestvar22 = bestvar2(2);
bestvar23 = bestvar2(3);

GAOptions_2=gaoptimset('PopulationSize',14,'Generations',1);
[bestvar3,bestobj3,history3,eval_count3]=ga(@Wing_Design,7,[],[],[],[],[bestvar21,bestvar22,bestvar23,0,0,0,0],[bestvar21,bestvar22,bestvar23,1,1,1,1],[],GAOptions);
count_3 = 120 - count_2 - eval_count3 - eval_count
options = optimset('MaxFunEvals',count_3);
[bestvar4,bestobj4]=fminsearchcon(@Wing_Design,bestvar3,[bestvar21,bestvar22,bestvar23,0,0,0,0],[bestvar21,bestvar22,bestvar23,1,1,1,1],[],[],[],options);
```



```matlab
%% Loop for reduce parameter pop and gen
% Initialize variables to store results
n_values = [];
mean_bestobj = [];
std_bestobj = [];
product_nm = [];

% Define range for n and m
max_n = floor(28 / 2); % Ensure condition 1 is satisfied
max_m = 27; % Max m to satisfy condition 1

% Iterate over different values of n and m
for n = 1:max_n
    for m = 1:max_m
        % Check if conditions are satisfied
        if (mod(n, 1) == 0) && (mod(m, 1) == 0) && (n > 0) && (m > 0) && (n * (m + 1) <= 28)
            % Initialize variables to store objective values
            bestobjs = zeros(100, 1);

            for iter = 1:100
                % Run genetic algorithm
                GAOptions = gaoptimset('PopulationSize', n, 'Generations', m);
                [bestvar, bestobj, ~, eval_count] = ga(@Test_Function3, 6, [], [], [], [], [0,0,0,0.5,0.5,0.5], [1,1,1,0.5,0.5,0.5], [], GAOptions);

                % Count remaining evaluations
                count_2 = 60 - eval_count;

                % Run constrained optimization
                options = optimset('MaxFunEvals', count_2);
                [bestvar2, bestobj2] = fminsearchcon(@Test_Function3, bestvar, [0,0,0,0.5,0.5,0.5], [1,1,1,0.5,0.5,0.5], [], [], [], options);

                % Store the best objective value
                bestobjs(iter) = bestobj2;


            end

            % Calculate mean and standard deviation of best objective values
            mean_best = mean(bestobjs);
            std_best = std(bestobjs);

            % Store n, mean, standard deviation, and product of n and (m + 1)
            n_values = [n_values, n];
            mean_bestobj = [mean_bestobj, mean_best];
            std_bestobj = [std_bestobj, std_best];
            product_nm = [product_nm, n * (m + 1)];
        end
    end
end

% Plot mean against product of n and (m + 1)
figure;
% yyaxis left
plot(product_nm, mean_bestobj, '.', 'MarkerSize', 14, 'Color','black');
% yyaxis right
% plot(product_nm, std_bestobj, '.', 'MarkerSize', 14, 'Color','red');
% ylim auto
ylim
xlabel('Total Runs of GA');
ylabel('Mean of Best Objective');
title('Mean of Best Objective aganist Total Runs of GA');
[min_mean, min_index] = min(mean_bestobj);
min_product = product_nm(min_index);

% Circle the lowest point
hold on;
plot(product_nm(min_index), mean_bestobj(min_index), 'ro', 'MarkerSize', 12); % Mark the lowest point with a red circle
line([min(product_nm), max(product_nm)], [min_mean, min_mean], 'Color', 'g', 'LineStyle', '--', 'LineWidth', 1.5); % Draw a green dashed line
```