Term Project Title: Reinforcement optimization of tunnel liner subjected to bending.

Submitted by:

Manas Bhadury

M.S. Research scholar

Roll No.-CE22S019

Course Title: Structural optimization (CE6730)

Course Instructor: Dr. Phanisri Pradeep Pratapa

Department of Civil Engineering

Indian Institute of Technology Madras

Chennai, India

Abstract: Recently fibre-reinforced concrete is being widely used in tunnel liners due to it's post peak load carrying capacity, capability of crack bridging and capability of resisting diffused stresses in the segment. However, it has been found out that using hybrid reinforcement i.e. steel rebar and steel fibre is much more beneficial from load carrying capacity and ductility viewpoint. In this study, an optimization has been proposed to determine the optimum dosages of steel rebar and steel fibre to get the maximum toughness under bending. A gradient and non-gradient based approaches have been compared in this context.

Problem statement: A tunnel liner of internal diameter 5.8 m and external diameter of 6.3 m is taken in this study. A line load is acting in the mid-section. The domain of the tunnel liner is shown in the figure1. We need to obtain the most optimal combination of rebar-fibre to get the maximum possible stiffness.

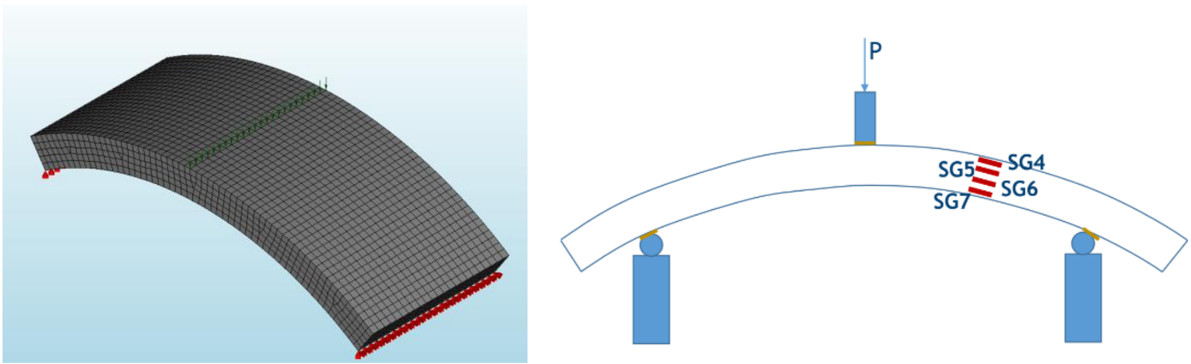

Figure 1: Schematic of the tunnel liner.

Objectives: The key objectives of the study are-

- To get the optimum dosages of rebar and fibre.
- To compare different optimization techniques to get the desired dosages.

Methodology: To get the toughness, we need to compute the area under the load-deflection curve. However, the complexity of the flexural response of the liner makes this task difficult. To simplify the calculation, two points have been chosen in the curve, one where the slope of the curve is changing largely (P1,u1), and another is corresponding to the maximum load(Pmax,umax) as shown in figure 2.

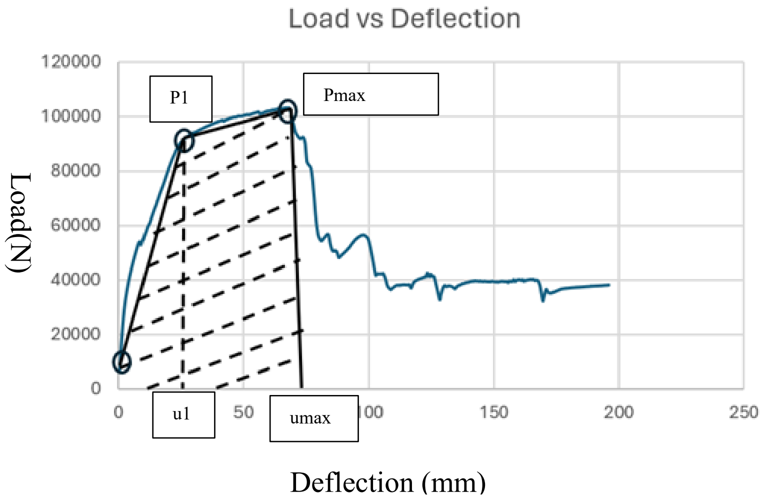

Figure 2: Methodology: simplification of area calculation

Derived expressions from finite element simulations and curve fitting:

$$P1 = -32 + 859.7y + 26.2x - 1010.7y^2 + 77.3xy - 9.7x^2$$

$$Pmax = -43.6459 + 145.0015y + 131.0431x - 218.7569y^2 + 22.4834xy - 15.5354x^2$$

$$\text{u1} = 18.6153 - 13.1276y + 8.8413x + 17.5066y^2 + 2.1654xy - 0.6867x^2$$

$$\text{umax} = 125.5877 - 24.6597y - 51.1271x - 36.5219y^2 + 27.0401xy + 6.6905x^2$$

$$Hence, Area\ under\ the\ curve = \frac{1}{2} * P1 * u1 + \frac{1}{2} * (P1 + Pmax) * (umax - u1)$$

<u>Optimization formulation</u>:

$$Minimize\ \ -A$$

Design variables: x, y

Constraints:

i)$1 \leq x \leq 3$

ii) $0.13 \leq y \leq 0.57$

x= dosage of steel rebar (in %)

y=dosage of fibre (in %)

Matlab code using fmincon (gradient based approach):

```
% Define x and y as symbolic variables
syms x y;

% Define expressions for P1, P2, u1, and u2
P1 = -32 + 859.7 * y + 26.2 * x - 1010.7 * y^2 + 77.3 * x * y - 9.7 * x^2;
P2 = -43.6459 + 145.0015 * y + 131.0431 * x - 218.7569 * y^2 + 22.4834 * x * y - 15.5354 *
x^2;
u1 = 18.6153 - 13.1276 * y + 8.8413 * x + 17.5066 * y^2 + 2.1654 * x * y - 0.6867 * x^2;
u2 = 125.5877 - 24.6597 * y - 51.1271 * x - 36.5219 * y^2 + 27.0401 * x * y + 6.6905 * x^2;

% Define the toughness expression
Toughness = 0.5 * P1 * u1 + 0.5 * (P1 + P2) * (u2 - u1);

% Convert the symbolic expression to a MATLAB function
Toughness_fun = matlabFunction(Toughness);

% Define initial guess
x0 = [2.8, 1];

% Define bounds for X
lb = [1, 0.13]; % Lower bounds
ub = [3, 1]; % Upper bounds

% Call fmincon
[X_opt, fval, exitflag, output] = fmincon(@(x) -Toughness_fun(x(1), x(2)), x0, [], [], [], [], lb,
ub);

% Display results
fprintf('Optimal X: %.4f, %.4f\n', X_opt(1), X_opt(2));
fprintf('Maximum toughness: %.4f\n', -fval);
disp(output);

% Plot the surface of the objective function
figure;
[X_mesh, Y_mesh] = meshgrid(linspace(1,4, 100), linspace(0, 1, 100));
surf(X_mesh, Y_mesh, Toughness_fun(X_mesh, Y_mesh));
xlabel('X');
ylabel('Y');
zlabel('Toughness');
title('Surface Plot of Toughness Objective Function');

figure;
contour(X_mesh, Y_mesh, Toughness_fun(X_mesh, Y_mesh), 'LevelStep', 200, 'LineWidth',
1.5);
```

```
hold on;
plot(x0(1), x0(2), 'ro', 'MarkerSize', 10, 'LineWidth', 1.5);  % Initial guess
plot(X_opt(1), X_opt(2), 'go', 'MarkerSize', 10, 'LineWidth', 1.5);  % Optimal point
xlabel('X', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('Y', 'FontSize', 12, 'FontWeight', 'bold');
title('Contours of Toughness Objective Function', 'FontSize', 14, 'FontWeight', 'bold');
legend('Contours', 'Initial Guess', 'Optimal Point', 'Location', 'northwest', 'FontSize', 10);
xlim([1, 4]);
ylim([0, 0.6]);
grid on;
box on;
set(gca, 'LineWidth', 1.5, 'FontSize', 10, 'FontWeight', 'bold');
```

Matlab code using genetic algorithm:

```
% Define x and y as symbolic variables
syms x y;

% Define expressions for P1, P2, u1, and u2
P1 = -32 + 859.7 * y + 26.2 * x - 1010.7 * y^2 + 77.3 * x * y - 9.7 * x^2;
P2 = -43.6459 + 145.0015 * y + 131.0431 * x - 218.7569 * y^2 + 22.4834 * x * y - 15.5354 *
x^2;
u1 = 18.6153 - 13.1276 * y + 8.8413 * x + 17.5066 * y^2 + 2.1654 * x * y - 0.6867 * x^2;
u2 = 125.5877 - 24.6597 * y - 51.1271 * x - 36.5219 * y^2 + 27.0401 * x * y + 6.6905 * x^2;

% Define the toughness expression
Toughness = 0.5 * P1 * u1 + 0.5 * (P1 + P2) * (u2 - u1);

% Convert the symbolic expression to a MATLAB function
Toughness_fun = matlabFunction(Toughness);

% Define the objective function
objective_function = @(x) -Toughness_fun(x(1), x(2));

% Define initial guess
x0 = [2 0.7];

% Define bounds for X
lb = [1, 0.13]; % Lower bounds
ub = [3, 1]; % Upper bounds

% Call the GA solver
options = optimoptions('ga', 'Display', 'iter');
[X_opt, fval, exitflag, output] = ga(objective_function, 2, [], [], [], [], lb, ub, [], options);

% Display results
fprintf('Optimal X: %.4f, %.4f\n', X_opt(1), X_opt(2));
fprintf('Maximum toughness: %.4f\n', -fval);
```

```
disp(output);

% Plot the surface of the objective function
figure;
[X_mesh, Y_mesh] = meshgrid(linspace(1, 4, 100), linspace(0, 1, 100));
surf(X_mesh, Y_mesh, Toughness_fun(X_mesh, Y_mesh));
xlabel('X');
ylabel('Y');
zlabel('Toughness');
title('Surface Plot of Toughness Objective Function');

figure;
contour(X_mesh, Y_mesh, Toughness_fun(X_mesh, Y_mesh), 'LevelStep', 200, 'LineWidth',
1.5);
hold on;
plot(x0(1), x0(2), 'ro', 'MarkerSize', 10, 'LineWidth', 1.5);  % Initial guess
plot(X_opt(1), X_opt(2), 'go', 'MarkerSize', 10, 'LineWidth', 1.5);  % Optimal point
xlabel('X', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('Y', 'FontSize', 12, 'FontWeight', 'bold');
title('Contours of Toughness Objective Function', 'FontSize', 14, 'FontWeight', 'bold');
legend('Contours', 'Initial Guess', 'Optimal Point', 'Location', 'northwest', 'FontSize', 10);
xlim([1, 4]);
ylim([0, 0.6]);
grid on;
box on;
set(gca, 'LineWidth', 1.5, 'FontSize', 10, 'FontWeight', 'bold');
```

<u>Results</u>:

Using gradient based approach fmincon:

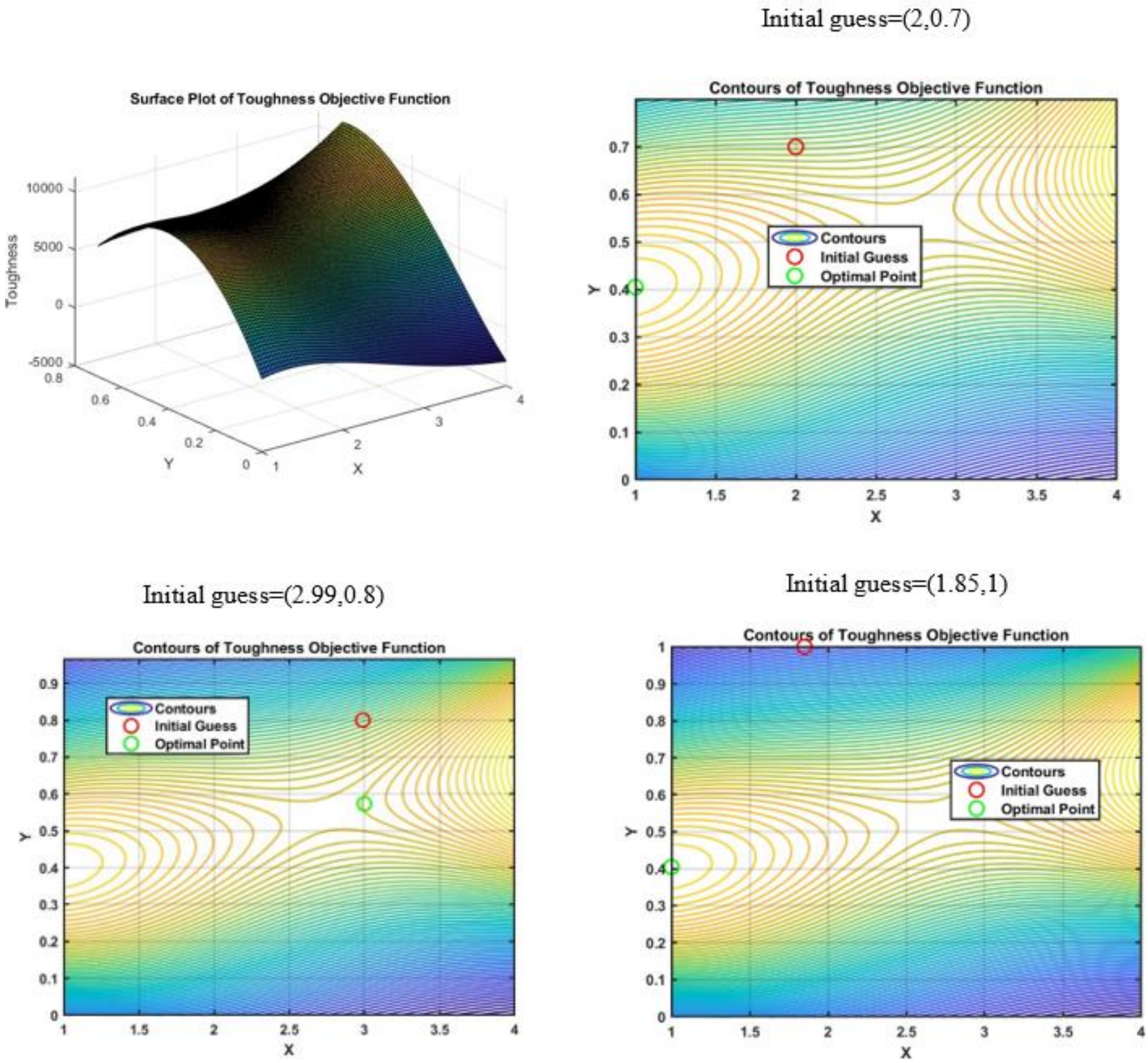

Figure 3: Results using fmincon in MATLAB(Surface and contour plots)

- From the gradient based approach, it has been evident that optimal solution changes as we change the initial guesses. If there are multiple local maxima and minima exist in a function, gradient based approach brings the initial guess to any of the local minima which does not guarantee its global optimality.
- In the Table1 it has been shown how the optimal point is changing depending on the initial guesses.

Table 1: Comparison for different initial guesses

Initial guess	Number of iterations	Objective function value	Optimal
2,0.7	8	10187.3378	1,0.4048
1.85,1	10	10187.3378	1,0.4048
2.99,0.8	7	8216.5927	3,0.5732

<u>Using genetic algorithm</u>: Genetic algorithm resolves the above-mentioned issue. Taking the same initial guesses, we get the optimal design variable. This is shown in Table2.

Table 2: Comparison for different initial guesses

Initial guess	Number of iterations	Objective function value	Optimal
2,0.7	71(generations)	10187.3378	1,0.4048
1.85,1	58(generations)	10187.3378	1,0.4048
2.99,0.8	75(generations	10187.3378	1,0.4048

-For initial guess (2.99,0.8), it has been evident from the contour plot that irrespective of the initial guess, the objective function value is not changing.

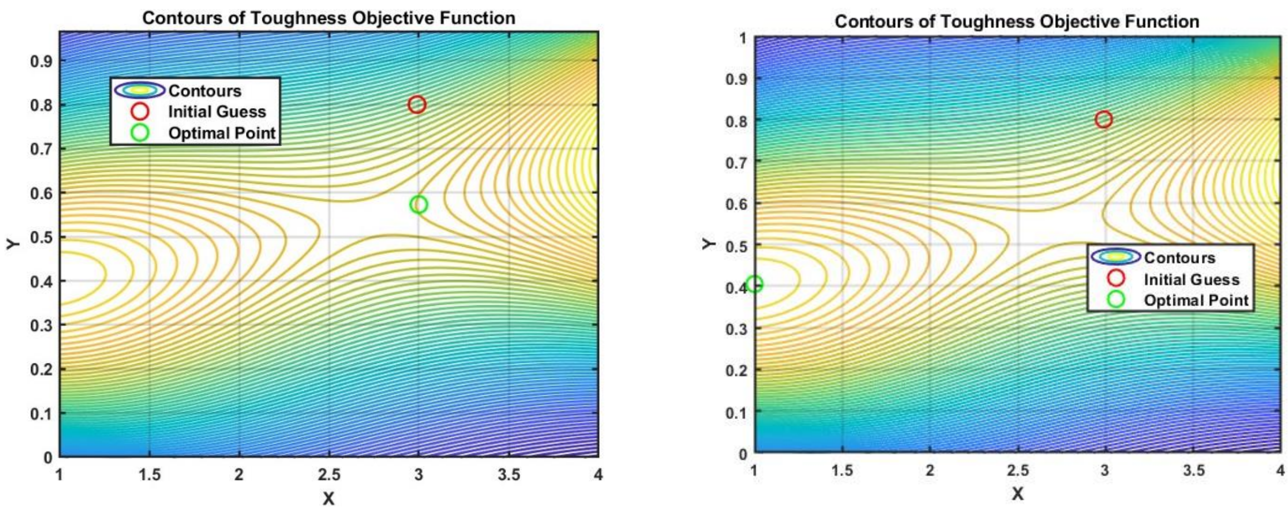

Figure 4: Contour plot comparison

Conclusion:

- As the percentage of steel increases ductility decreases.
- As the fibre increases ductility increase however after some point, increasing fibre has detrimental effects. Hence it decreases.
- So, this proposed optimization methodology can identify the optimum reinforcement for which toughness is maximum.
- Gradient based approach is not suitable as there are multiple local Maximas. Hence genetic algorithm is used, however genetic algorithm found out to be 10 times slower than gradient based algorithms.

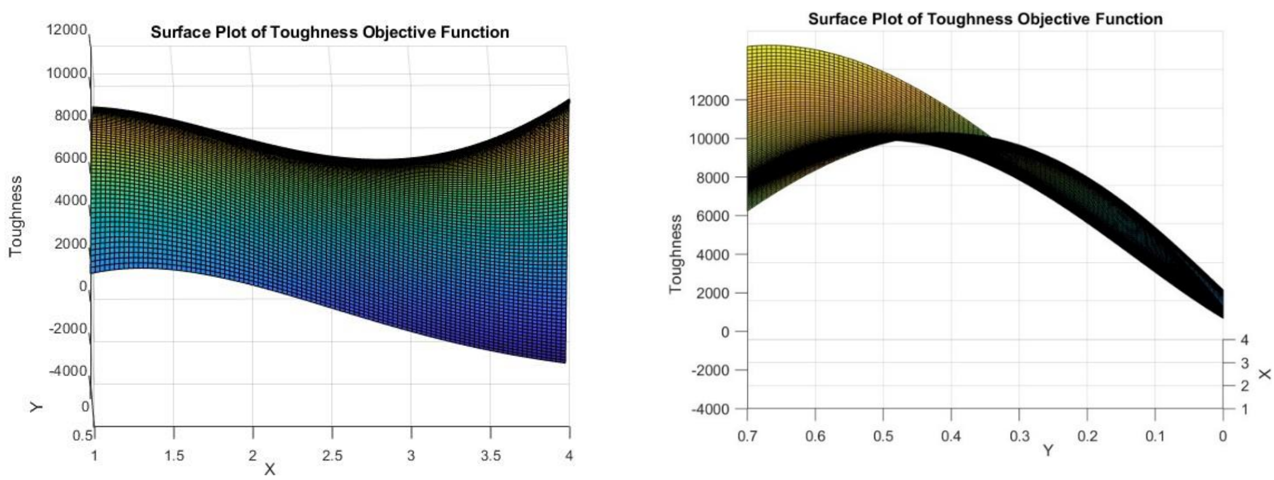

Figure 5: Variation of toughness with i) steel rebar and ii) steel fibre(Left to right)