Lecture - 08
MULTI-OBJECTIVE OPTIMIZATION

Reference: Book Chapter 6

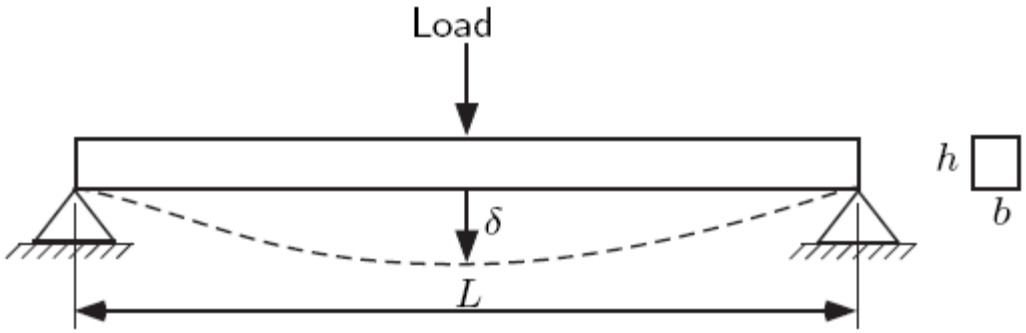
MULTI-OBJECTIVE PROBLEMS

- Many real-life design problems contains more than one design objective (to be maximized or minimized).
  - Example: Car design: Maximize Fuel Efficiency and Minimize Cost.
- Generally, in practical problems where multiple objectives need to be considered, the objectives tend to be conflicting in nature
  - Example: More fuel efficient cars (e.g., hybrid cars or cars with regenerative braking) tend to be more expensive than less efficient counterparts).
- *Multi-objective optimization is a methodical way to solve problems involving* ***multiple design objectives simultaneously***.

MULTI-OBJECTIVE PROBLEM EXAMPLE

Design a pinned-pinned beam, with the following given:

- Fixed beam features: its length *L* and width *b*, the material, and the load acting on it.
- The only parameter allowed to change is the height, *h*.
- Minimize the bending stress in the beam, and at the same time minimize the deflection at mid-span.



Design objectives:

1.μ_1 = Bending stress

2.μ_2 = Deflection

Design variable:

1.x_1 = Height

MULTI-OBJECTIVE PROBLEM STATEMENT

$$\min_{x} \quad [\mu_1(x) \quad \mu_2(x) \quad \dots \quad \mu_n(x)]^T$$

Multiple **objective functions**

subject to

$$g(x) \leq 0$$

Vector of **inequality constraints**

$$h(x) = 0$$

Vector of **equality constraints**

$$x_l \leq x \leq x_u$$

Side constraints

When *n* = 2, we call it a bi-objective problem.

CONCEPT OF PARETO SOLUTIONS

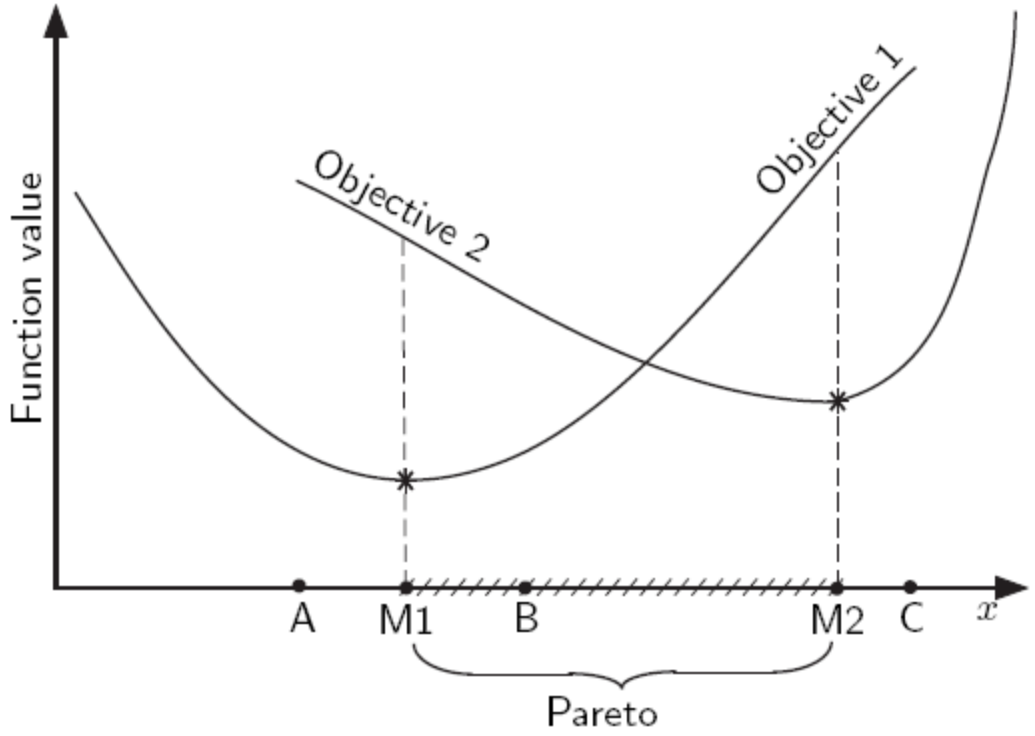
- One of the interesting features of multi-objective optimization is that the solution to the problem is generally **not unique**.
- A set of solutions called **Pareto Optimal Solutions** or the **best tradeoff solutions** form the complete solution set of the optimization problem.

Example

- Let's consider an unconstrained problem involving two design objectives, μ_1 and μ_2, which are functions of a single design variable x.
- We are interested in minimizing both design objectives simultaneously.

CONCEPT OF PARETO SOLUTIONS (CONT.)

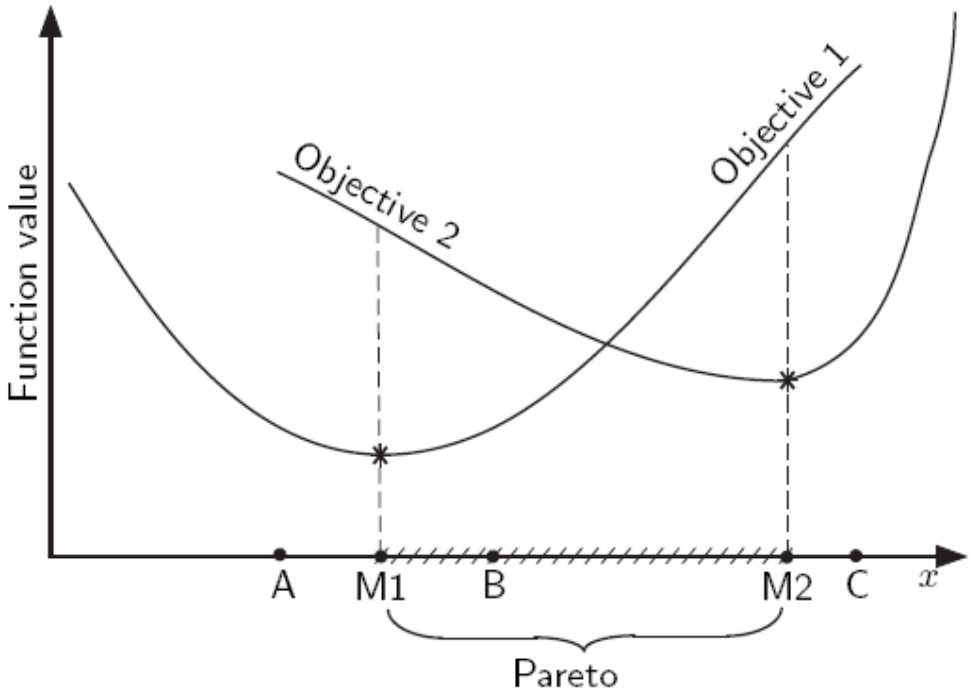
- If we minimize each objective function independently, ignoring the other objective, we will generally obtain the point that corresponds to the minimum of the objective being minimized.
- These two points, called anchor points, are indicated by star symbols.



CONCEPT OF PARETO SOLUTIONS (CONT.)

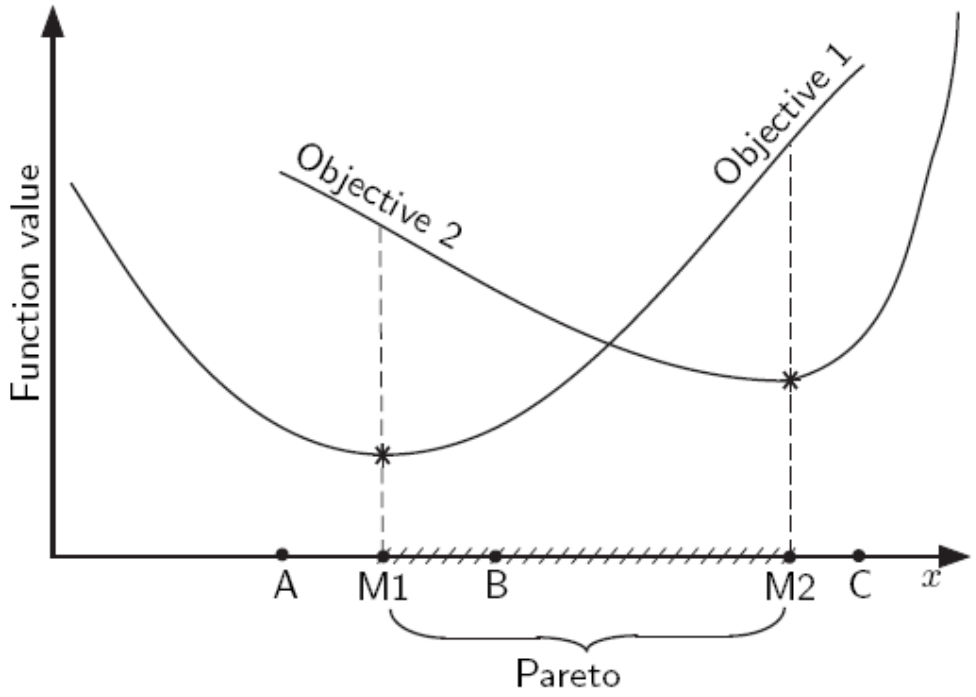
- When you move from the minimum of objective 1 (M1) to its right, say to a point B – Objective 2 decreases, but objective 1 increases.
- To improve μ_1, we have to compromise on the performance of μ_2, and vice versa. This is true for all the points between M1 and M2.
- Such points (between M1 and M2) are called **Pareto optimal solutions**, or **non-dominated solutions**.

Definition: *Pareto optimal solutions are those for which any improvement in one objective will result in the worsening of at least one other objective.*



CONCEPT OF PARETO SOLUTIONS (CONT.)

- If you move to the right of M2 to C or to left of M1 to A, both objectives increase simultaneously.
- These points beyond M1–M2 (e.g., A and C) are called **non-Pareto** or **dominated solutions**.

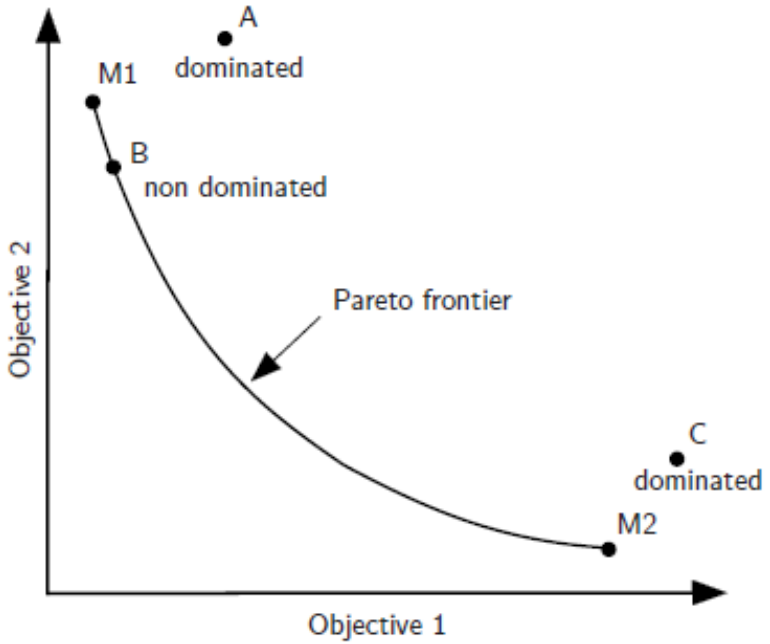


THE PARETO FRONTIER

- In the previous slides, Pareto solutions are identified in the design variable space, that is, x space.
- **Design objective space** is a plot, where each design objective is plotted on one axis – up to three objectives.
- The Pareto solutions plotted in the design objective space, mark a curve or a surface that is called the **Pareto frontier**.

PARETO FRONTIER: OBJECTIVE SPACE

- M1 and M2 form the end points of the Pareto frontier. Point M1 is where objective 1 has the least value, while M2 is where objective 2 has the least value.
- A and C, dominated points, do not lie on the Pareto frontier.
- There are tradeoffs associated with each Pareto point.



OBTAINING PARETO SOLUTIONS

- Many optimization algorithms (including most conventional or classical algorithms) are suited to solve single objective problems.
- One way to solve a multi-objective problem, by leveraging typical single-objective algorithms is to combine all the objectives into a single **Aggregate Objective Function**.
- If the single objective is optimized, a Pareto solution is obtained.

AGGREGATE OBJECTIVE FUNCTION

- Definition: An **Aggregate Objective Function** (**AOF**) is a combined function of all the design objectives.
- The AOF typically contains weight parameters to be selected by the designer. These parameters generally reflect the relative importance of each design objective.
- The type of final solution that you obtain will be directly dependent on the type of AOF that you use.
- A general guideline is to use an AOF that allows a designer to impose his/her design objective preferences in an unambiguous manner.

AGGREGATE OBJECTIVE FUNCTION FORMULATIONS

Some popular AOF formulations:

1. The weighted sum method
2. Compromise programming
3. Goal programming
4. Physical programming
5. Matlab optimization toolbox

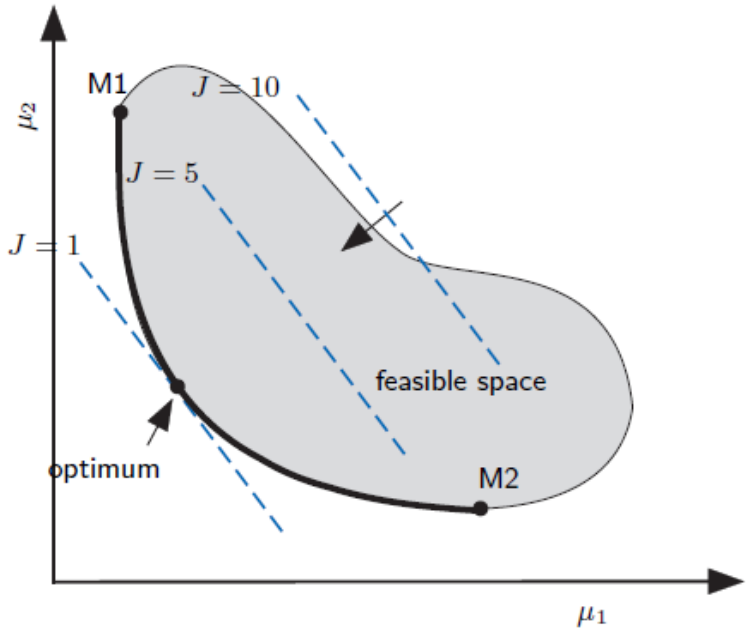
THE WEIGHTED SUM METHOD

- Aggregate Objective: A **weighted linear combination** of all the objective functions, $J(x)$.
- Each combination of weights corresponds to parallel straight lines of same slope for a bi-objective problem, or *n*-D hyperplanes for a *n*-objective problem.
- Minimizing $J(x)$ for different combination of weights ($1 \geq w \geq 0$) produces the different Pareto points on the Pareto frontier.

$$\min_{x} \quad J(x) = w_1\mu_1(x) + w_2\mu_2(x)$$

subject to

$$g(x) \leq 0$$
$$h(x) = 0$$
$$x_l \leq x \leq x_u$$

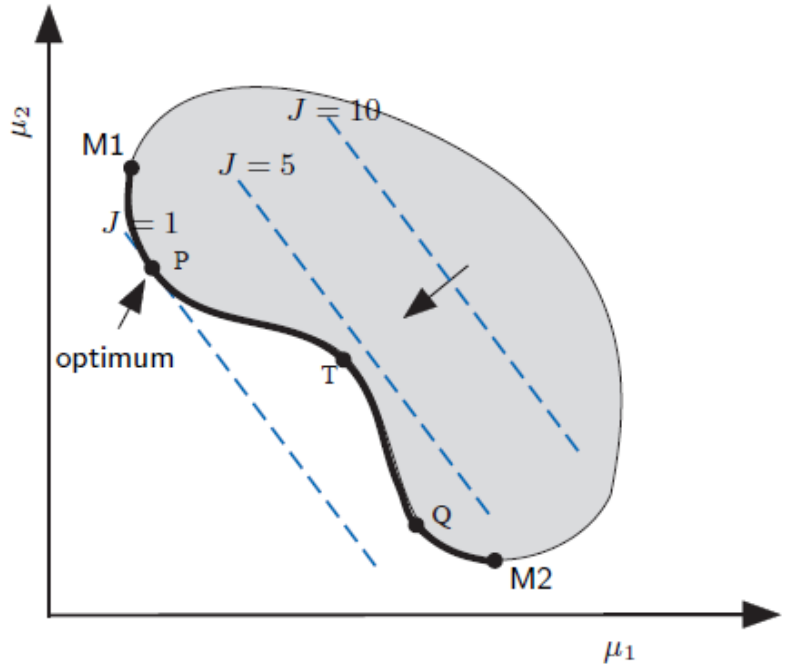


THE WEIGHTED SUM METHOD: STEPS

- As the optimization process decreases the value of J (for a particular weight combination), the solution approaches the Pareto frontier.
- If $w_1 = 0$, the constant value curves of J are parallel to μ_1, and the optimum point obtained will be M_2, the minimum of μ_2.
- Similarly, if $w_2 = 0$, you will get M_1 as the minimum.
- For all other combinations of w_1 and w_2, different points on the Pareto frontier are obtained.
- A typical way to set the weights is to uniformly vary them between 0 and 1, such that their sum is equal to 1.
- One of the deficiencies of the weighted sum method is that there is no easy way to know what values of w to use.

THE WEIGHTED SUM METHOD

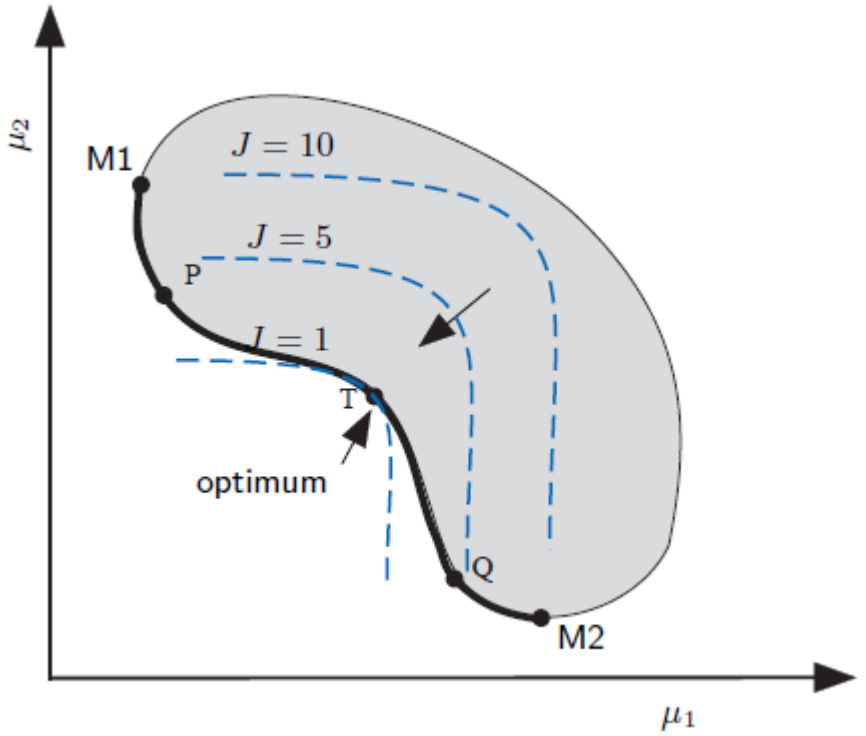
- The Weighted Sum Method may not be effective for **non-convex** Pareto frontiers.
- **Example**: The weighted sum method cannot obtain the Pareto points which lie in the non-convex region of the shown Pareto frontier (all points between P and Q).
- During minimization, the dashed line continues beyond point T, and gives you point P as the optimum solution.



COMPROMISE PROGRAMMING

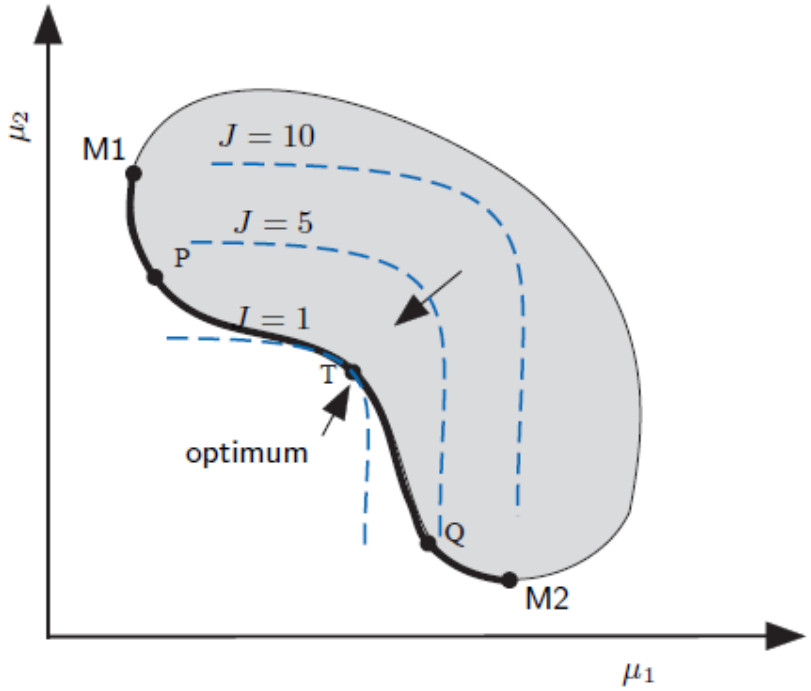
- Compromise Programming is a more effective extension of the weighted sum strategy.
- The objective functions have an exponent, *n*. The constant value curves of *J* are not straight lines. Instead, they are ***n*-th order curves**.

$$J(x) = w_1\mu_1(x)^n + w_2\mu_2(x)^n$$



COMPROMISE PROGRAMMING: STEPS

- Choose a high enough value for *n*, such that it reaches into the non-convex portions of Pareto frontier.
- As a general guideline, choose *n* as an even integer less than or equal to 8.
- In most cases, *n* = 2 or 4 yield satisfactory results.
- *n* = 2: weighted square sum method



GENERATING PARETO WITH MATLAB: CODE

Example

$$\min_{x_1,x_2} \quad \{\mu_1 = x_1^2 + 4x_2, \mu_2 = x_2^2 + 2\}$$

subject to

$$2x_1 + 3x_2^2 - 8 \leq 0$$

$$x_1 + x_2 - \frac{7}{2} = 0$$

$$0 \leq x_1 \leq 10$$

$$0 \leq x_2 \leq 5$$

GENERATING PARETO WITH MATLAB: CODE

Main file: **main.m**

```
% Initial Guess
x0 = [1 1];

% Lower and upper bounds
LB = [0 0];
UB = [10 5];

% Initialize counter j
j = 1;

% for loop that changes the weights. Each iteration of the for loop produces one Pareto point
for i = 0:0.02:1

% Assigning weights
    w1 = i;
    w2 = 1-i;

% Calling fmincon and passing weights to objfun and confun
[x,fval] = fmincon(@objfun,x0,[],[],[],[],LB,UB,@confun,[],w1,w2);

    % Finding objective functions values
    solx(j) = x(1)^2+4*x(2);
    soly(j) = x(2)^2+2;

    % Incrementing the counter
    j = j+1;
end

% Plotting the Pareto frontier
plot(solx,soly,'*')
xlabel('objective 1')
xlabel('objective 2')
```

c, ceq

f

x, w1, w2

x, w1, w2

```
function [c,ceq] = confun(x,w1,w2)

% Defining inequality constraints
c(1) = [2*x(1)+3*x(2)^2-8];

% Defining equality constraints
ceq = [x(1)+x(2)-(7/2)];
```

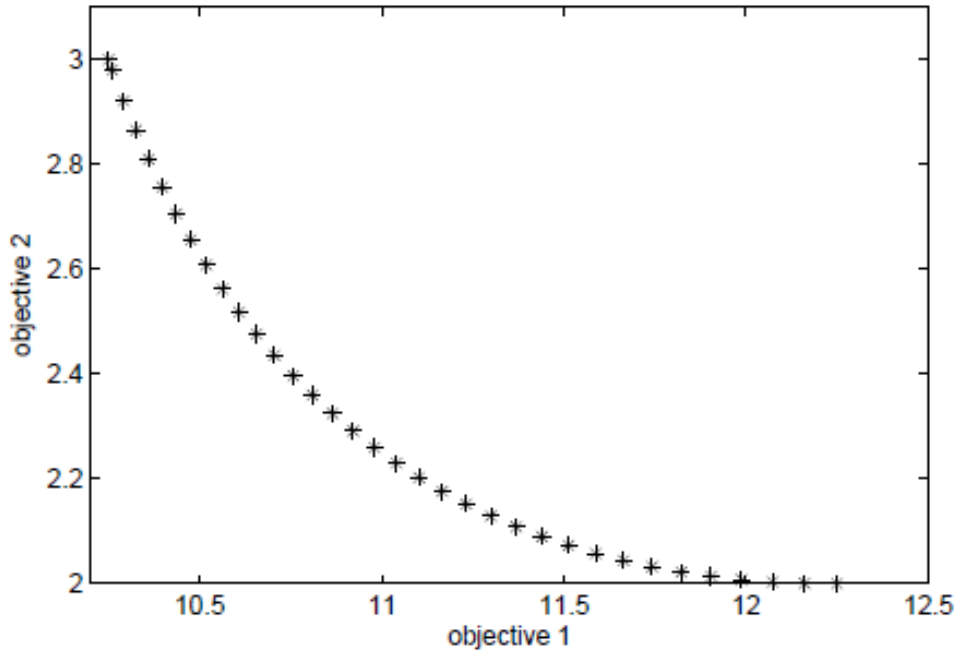
Constraint file: **confun.m**

```
function [f] = objfun(x,w1,w2)

% Defining mu1 and mu2
f1 = x(1)^2+4*x(2);
f2 = x(2)^2+2;

% Defining the AOF
f = w1*f1+w2*f2;
```

Objective function file: **objfun.m**

GENERATING PARETO WITH MATLAB: RESULTS

- The objective values of the Pareto solutions (solx, soly), are plotted to obtain the Pareto frontier.
- The number of Pareto points to be generated is set by the rate of variation of the weights. More division of weights between 0 and 1 generally yields more Pareto points, but also demands greater computational time.



GOAL PROGRAMMING

- Instead of minimizing design objectives, we might wish to reach a given target value or goal for each objective.
- For example, the stress design objective (μ_1) is required to be as close to 1,500 MPa as possible, while the value of deflection (μ_2) is required to be as close to 2 inches as possible.
- Goal programming AOF can be formulated using the compromise programming concept, where *n* is an even integer (often 2), e.g.:

$$J(x) = w_1 (\mu_1 - 1,500)^n + w_2 (\mu_2 - 2)^n$$

- The smallest theoretical value of *J* is zero, when μ_1 = 1500 and μ_2 = 2. In most real-life problems, this point will not be achievable.
- By choosing an appropriate set of weights (w_1 and w_2) for the design objectives, we can obtain a Pareto solution that reflects the best trade-offs.

GOAL PROGRAMMING

- A normalized version of the goal programming formulation

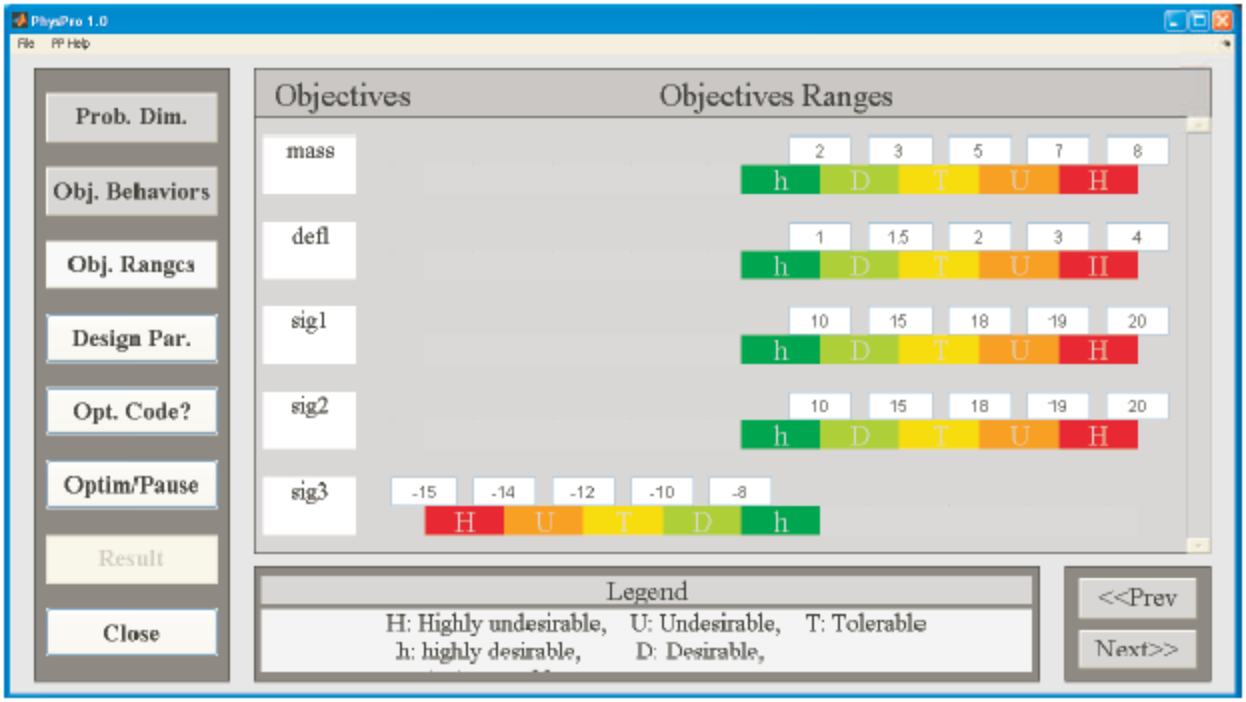
$$J(x) = w_1 \left(\frac{\mu_1 - \overline{\mu}_{1g}}{\overline{\mu}_{1b} - \overline{\mu}_{1g}} \right)^n + w_2 \left(\frac{\mu_2 - \overline{\mu}_{2g}}{\overline{\mu}_{2b} - \overline{\mu}_{2g}} \right)^n$$

- μ_{1g} and μ_{2g} are the goal values of the objectives μ_1 and μ_2, respectively. μ_{1b} and μ_{2b} are reference bad values of the objectives μ_1 and μ_2, respectively.
- We set μ_{1g} = 1, 500 and μ_{2g} = 2; and we might have μ_{1b} = 8, 000 and μ_{2b} = 10.
- Such a normalized formulation will have more stable numerical behavior, especially when the objectives have different orders of magnitude.
- Conceptually, these formulations can be readily extended to cases of multiple objectives.

EXPRESSING A PREFERENCE – PHYSICAL PROGRAMMING

- The **physical programming** method provides a more realistic expression of preference.
- Different ranges of differing desirability for the objectives.

Highly desirable	Desirable	Tolerable	Undesirable	Highly Undesirable



MULTIOBJECTIVE OPTIMIZATION: MATLAB TOOLBOX

goal attainment and **minimax**

- **fgoalattain**: The goal attainment Matlab function involves reducing the value of a linear or nonlinear vector function in order to attain the goal values given in a goal vector.
  - A weight vector is used, which is intended to indicate the relative importance of the goals.
  - The goal attainment problem may also be subject to linear and nonlinear constraints.
- **fminimax**: The minimax problem involves minimizing the worst-case value of a set of multivariate functions, possibly subject to linear and nonlinear constraints.

EXAMPLE: USING MATLAB

Consider the following bi-objective optimization problem.

$$\mu_1 = \sin\theta$$
$$\mu_2 = 1 - \sin^7\theta$$
$$0.5326 \leq \theta \leq 1.2532$$

(a) Obtain several optimal points on the Pareto frontier using the weighted sum method. Use MATLAB's fmincon function for optimization. Plot the points in the μ_1-μ_2 space.

(b) Do you think that the weighted sum method performs satisfactorily in obtaining points on the Pareto frontier?

(c) Use compromise programming with an appropriate value for the exponent to potentially obtain a better Pareto frontier.

EXAMPLE: WEIGHTED SUM CODE

(a) The weighted sum method

main function

```
clear; clc;
x0 = 0;
LB = 0.5326; UB = 1.2532;
i = 1;
for alpha = 0:0.1:1
   xopt(i) = fmincon('fun32ws', x0, [], [], [], [], LB, UB, 'nonlcon3_2', [], alpha);
   plotmu1(i) = sin(xopt(i));
   plotmu2(i) = 1-(sin(xopt(i)))^7;
   x0 = xopt(i);
   i = i+1;
end
plot(plotmu1, plotmu2, 'b*');  xlabel('mu1');  ylabel('mu2');
```

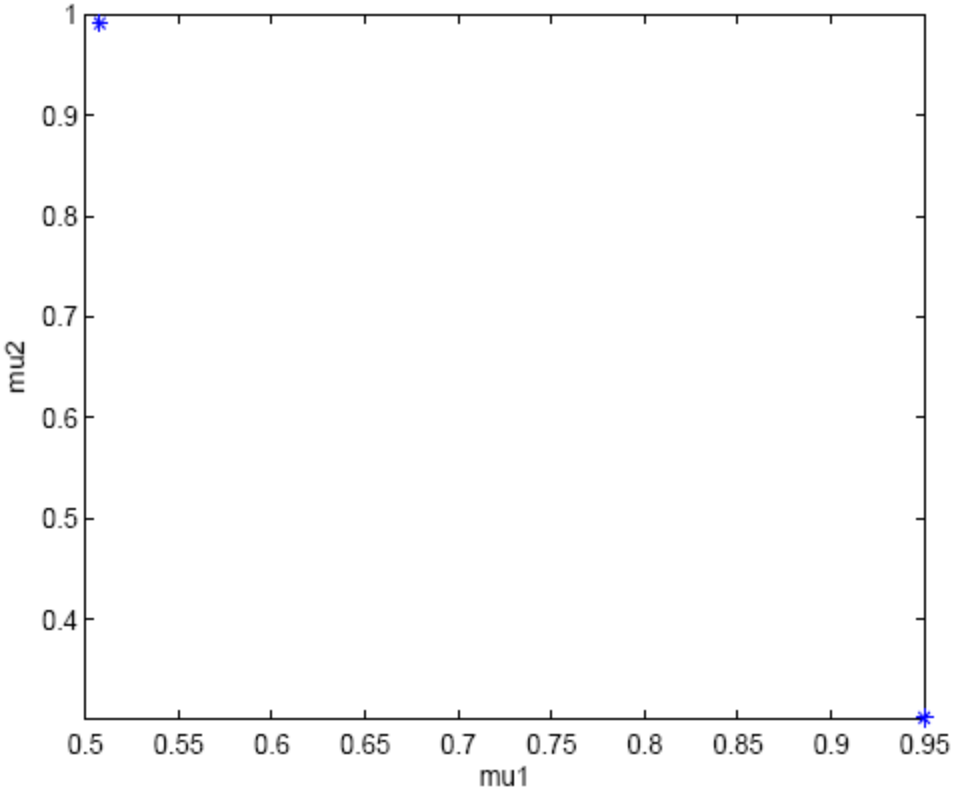
objective function

```
function f = fun32ws(x, alpha)

f = alpha*sin(x) + (1-alpha)*(1-(sin(x))^7);
```

constraint function

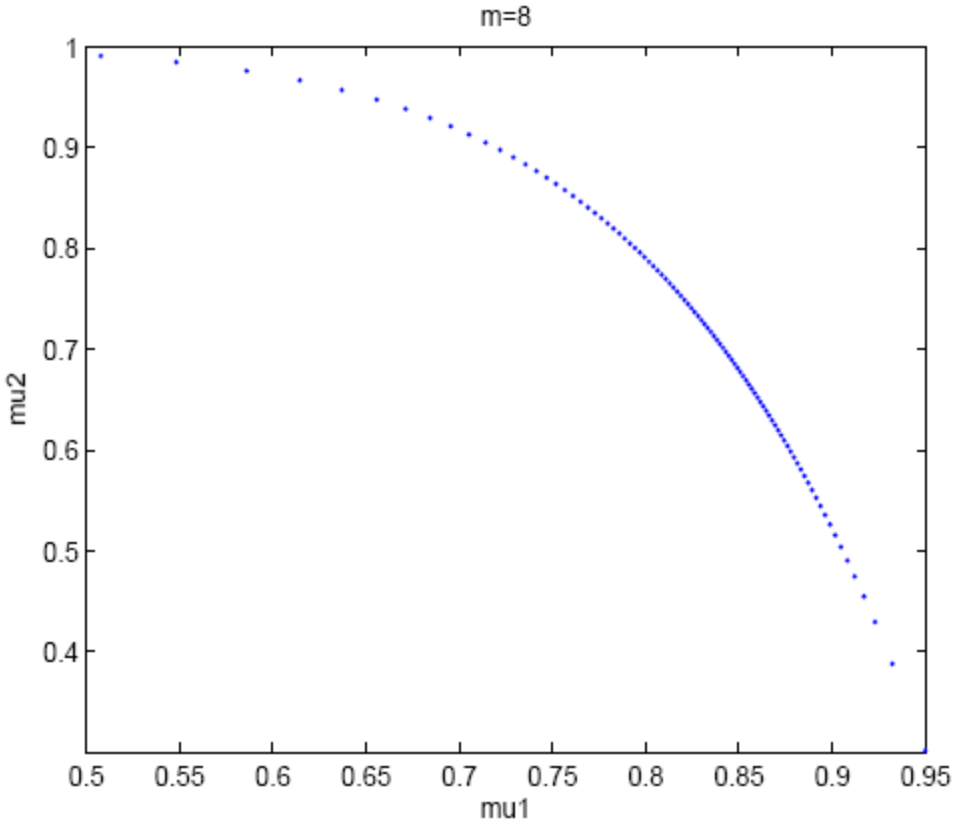
```
function [C, Ceq] = nonlcon3_2(x, alpha)

C = [];

Ceq = [];
```

EXAMPLE: WEIGHTED SUM RESULTS



The Weighed Sum method performs very poorly in obtaining the Pareto frontier.

EXAMPLE: COMPROMISE PROGRAMMING RESULTS



Compromise programming produces a satisfactory Pareto frontier.

3-OBJECTIVE EXAMPLE PROBLEM

$$\min_x \{\mu_1, \mu_2, \mu_3\}$$

subject to

$$(\mu_1 - 1)^2 + (\mu_2 - 1)^2 + (\mu_3 - 1)^2 \leq 1$$

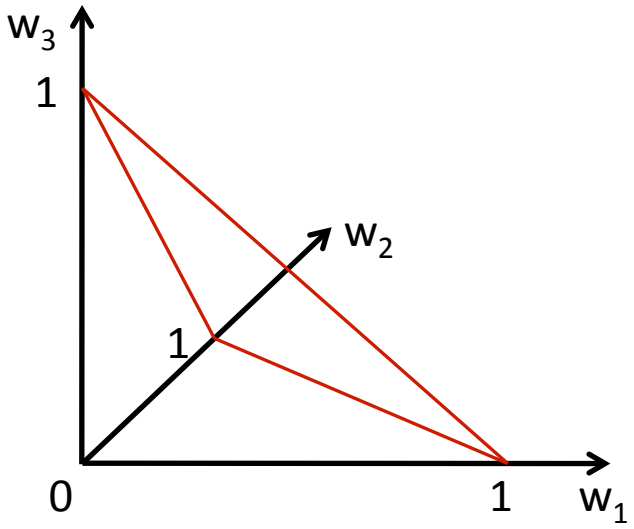
$$\mu_1 = x_1$$

$$\mu_2 = x_2$$

$$\mu_3 = x_3$$

Using the weighted sum method, plot the Pareto frontier for the above problem.(Hint: You will need three weights, each corresponding to a design objective. Allow each of these weights to vary between 0 and 1 when executing your code.)

3-OBJECTIVE PROBLEM: WEIGHT ASSIGNMENT

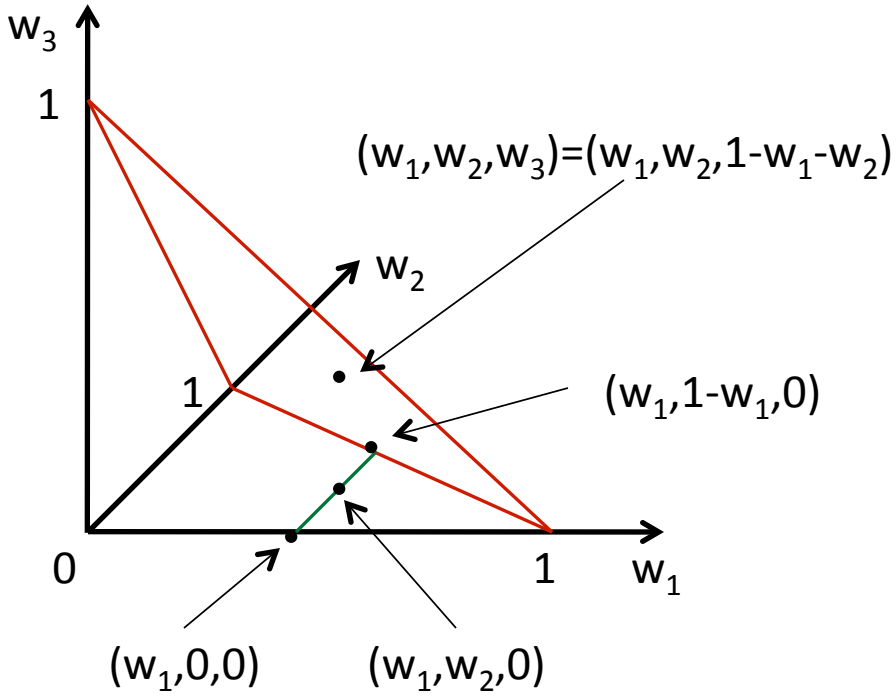


- Any weight ratio $h_1:h_2:h_3$ ($h_1 \geq 0, h_2 \geq 0, h_3 \geq 0$, and $h_1+h_2+h_3>0$) can be converted to

$$w_1 : w_2 : w_3 = \frac{h_1}{h_1+h_2+h_3} : \frac{h_2}{h_1+h_2+h_3} : \frac{h_3}{h_1+h_2+h_3}$$

- w_1, w_2, and w_3 satisfy $w_1+w_2+w_3=1$ $(0 \leq w_1, w_2, w_3 \leq 1)$
- (w_1, w_2, w_3) is on the plane surrounded by the red lines
- Use all the points on the plane as weight ratios.

Computationally Efficient Solution




```
Two loops for all points
w1+w2+w3=1

 for i=0:0.02:1
   for j=0:0.02:1-i
      w1=i;w2=j;w3=1-i-j;
            ……
   end
end
```

- On plane w_1-w_2, a point on the red line can be $(w_1,1-w_1,0)$.
- w_2 of a point on the plane surrounded by red lines can be projected to plane w_1-w_2.
- w_2 of a point on the green line is between 0 and $1-w_1$.
- w_3 of a point on the plane surrounded by red lines is $1-w_1-w_2$.

PARETO FOR 3-OBJECTIVE PROBLEM

3D Pareto