Multi-Objective Optimization

Concepts of Multi-Objective Optimization

In many engineering problems, we need to minimize or maximize two or more conflicting Objectives

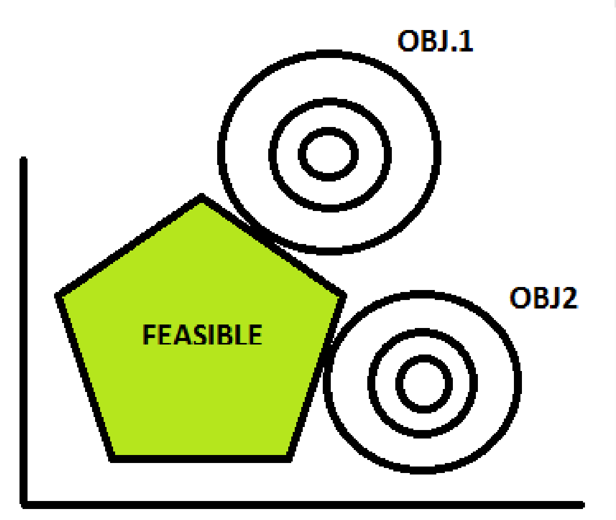
Eg. A) Minimize structural weight and maximize stiffness.

b) Minimize cost and maximize strength.

These objectives are conflicting. The optima of these objective functions will not coincide at the same point.

Multi-Objective Optimization addresses this problem.

How to find a solution which satisfies both objective functions.

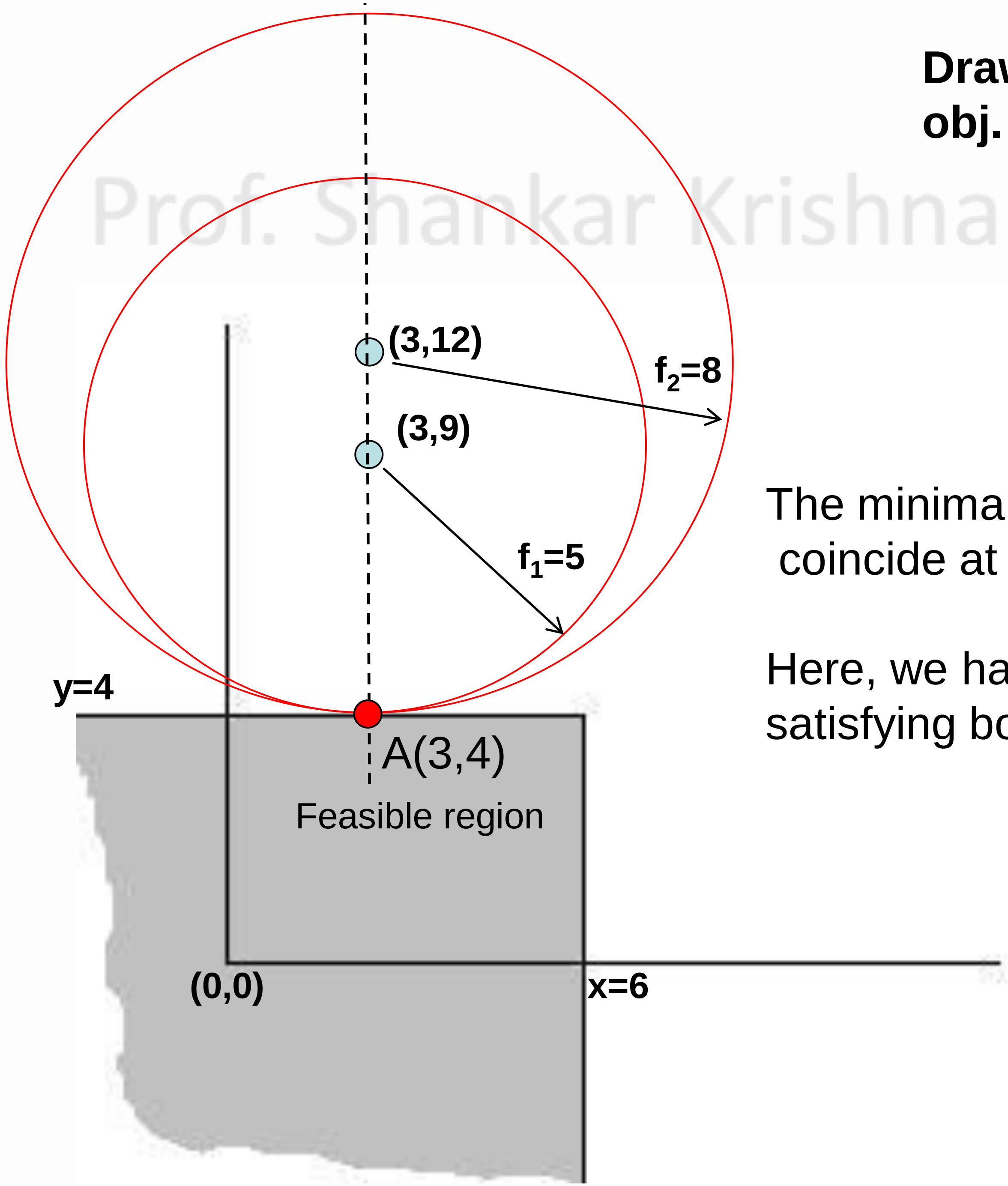


Consider the Minimization of two Objective functions:

$$\textit{Minimize} \quad f_1 = \sqrt{(x-3)^2 + (y-9)^2}$$ Circular contours centered at (3,9)

$$f_2 = \sqrt{(x-3)^2 + (y-12)^2}$$ Circular contours centered at (3,12)

$$\textit{subject to} \quad x \le 6;$$
$$y \le 4;$$



Draw contours of two obj. functions.

$f_{1.min} = 5$

$f_{2.min} = 8$

The minima of both functions coincide at A(3,4).

Here, we have one unique solution satisfying both Objective functions.

Now let us change f_2:

$$\text{Minimize} \quad f_1 = \sqrt{(x-3)^2 + (y-9)^2}$$

Circular contours centered at (3,9)

$$\boxed{f_2 = \sqrt{(x-10)^2 + (y-1)^2}}$$

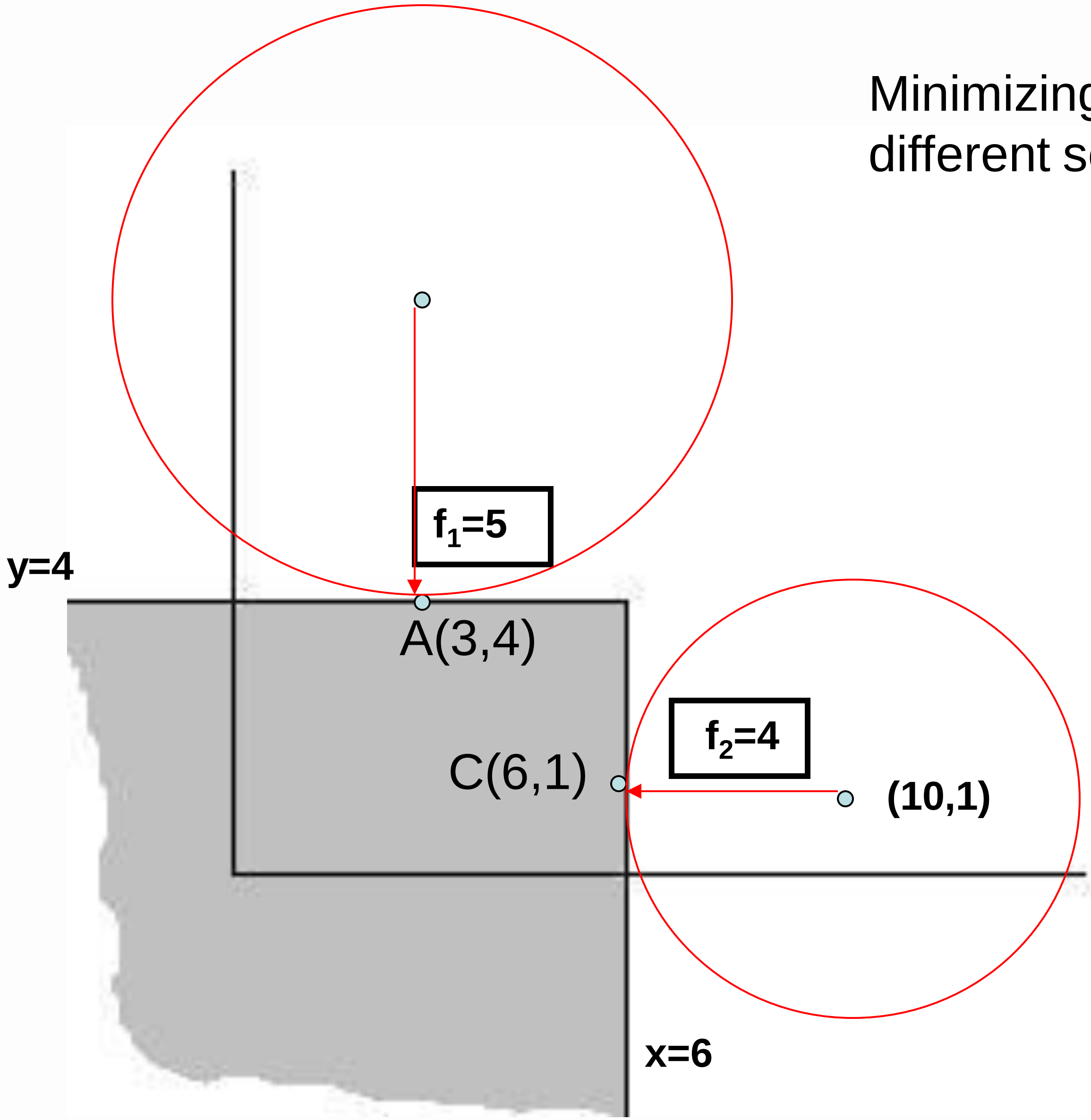
Circular contours centered at (10,1)

$$\text{subject to} \quad x \le 6;$$

$$y \le 4;$$

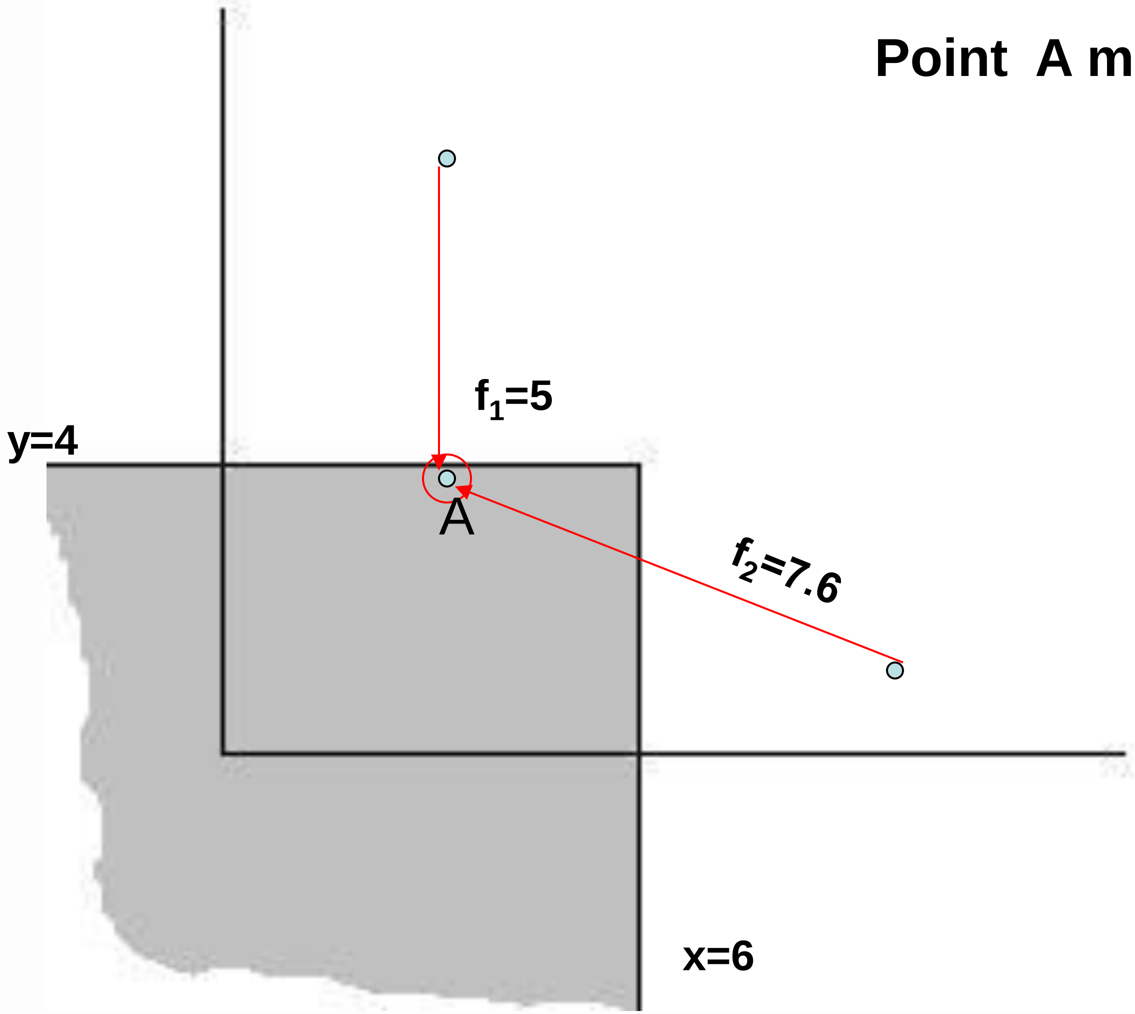
Prof. Shankar Krishnapillai
IIT Madras

Minimizing f_1 and f_2 give two different solutions A and C.

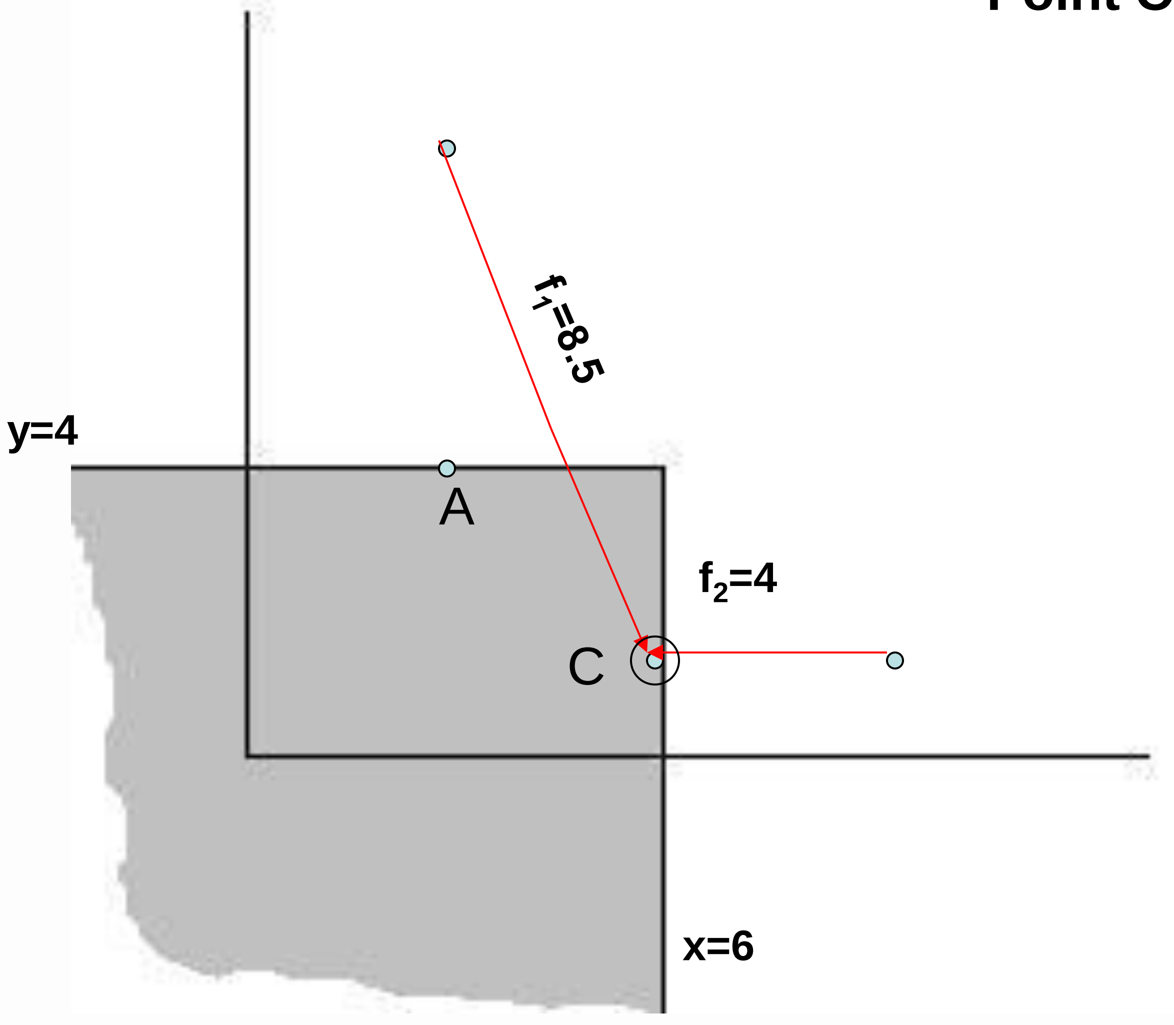


We get two points A and C .

Point A minimizes f_1, not f_2

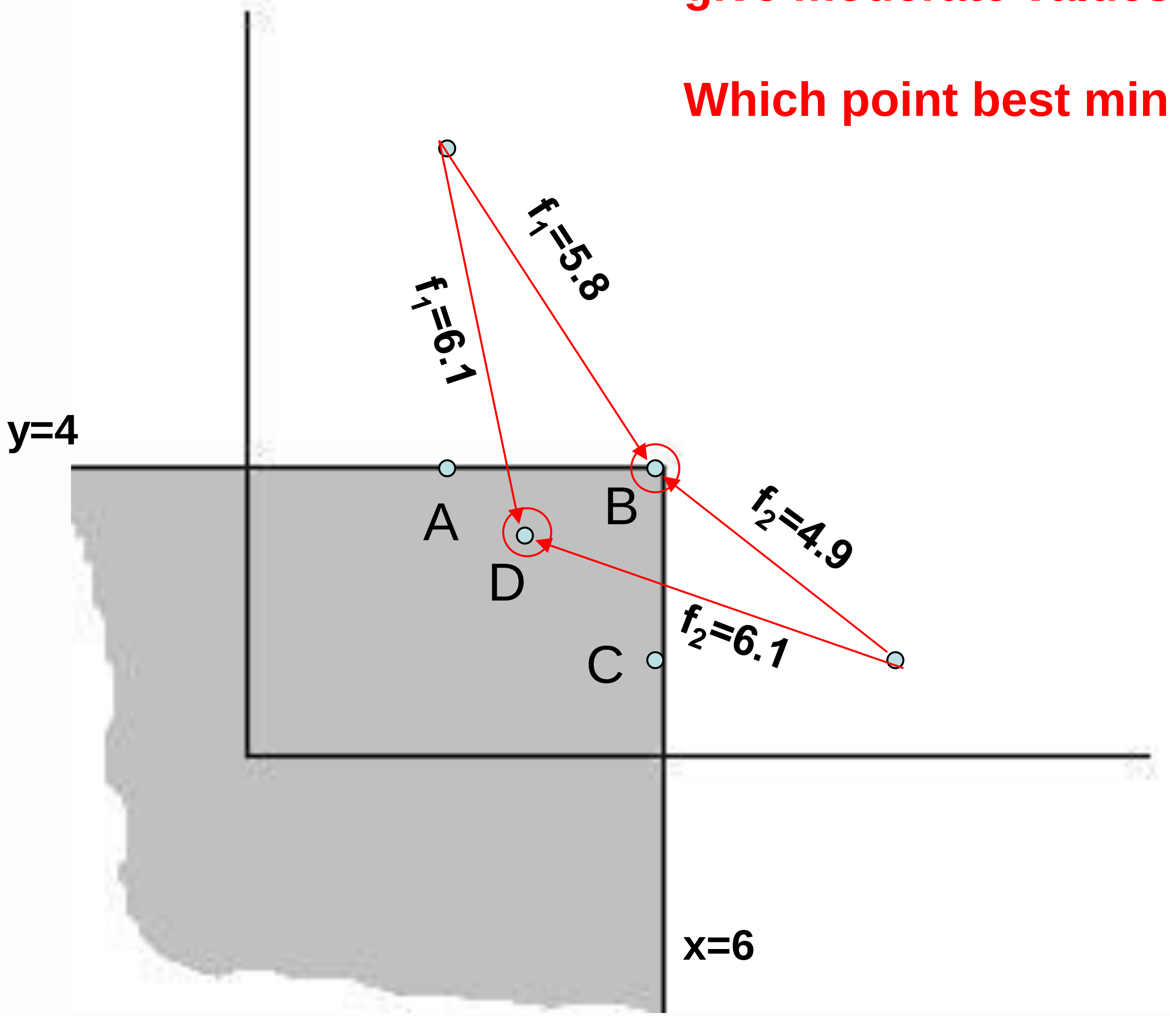


Point C minimizes f_2, not f_1



We can find other points B, D which give moderate values for f_1 and f_2.

Which point best minimizes both f_1 and f_2 ?



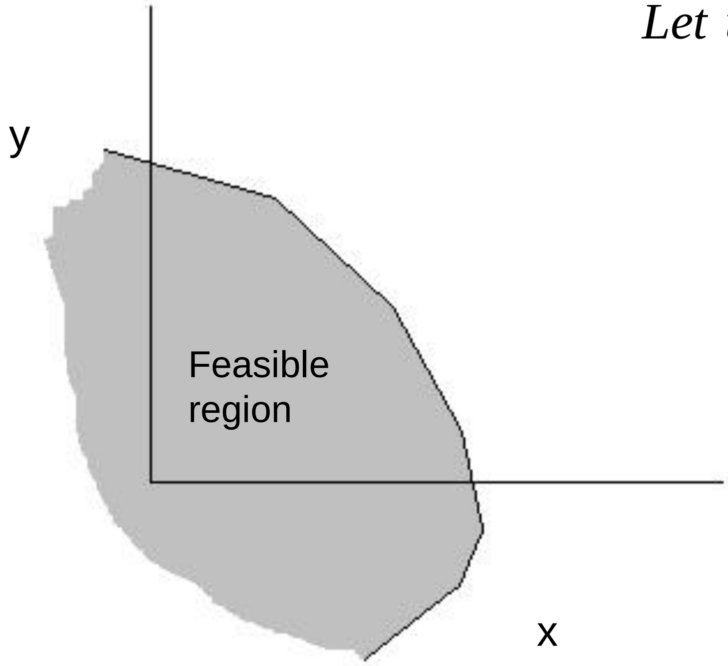
The concept of Dominated and Non-Dominated points are designed to address such questions.

Consider the standard optimization problem

Min $f_1(x,y)$, $f_2(x,y)$

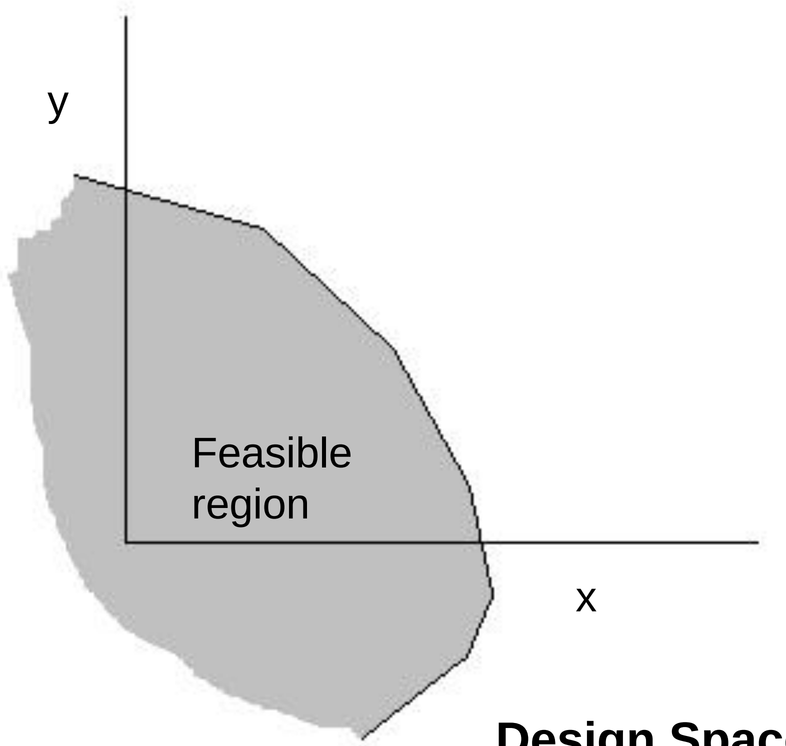
$$subject\ \ to:$$
$$g_1 \geq 0;$$
$$h_1 = 0;$$

Let us draw the feasible region

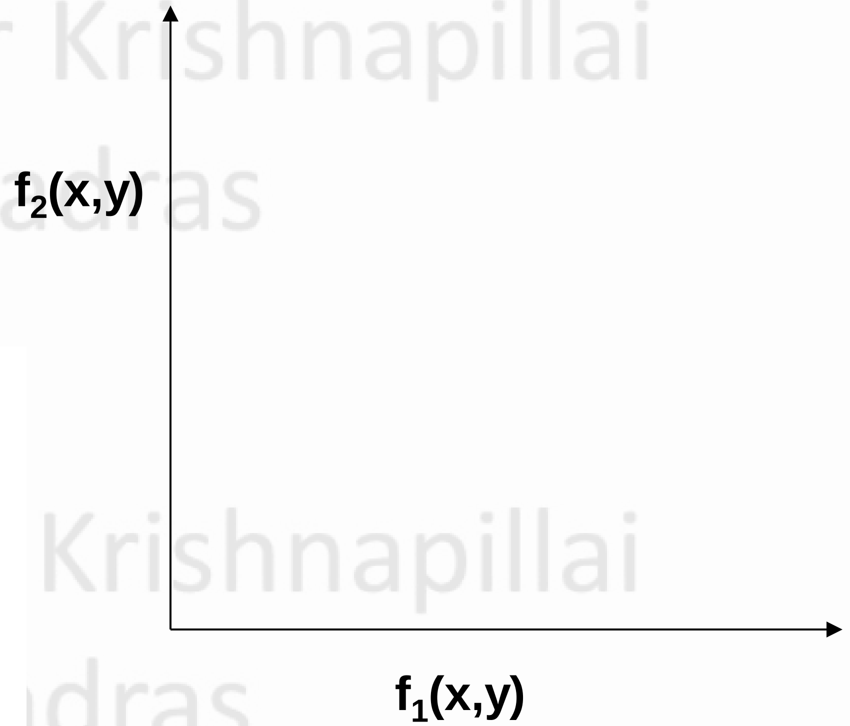


Design Space

The feasible points in the Design Space are mapped onto the Criterion Space

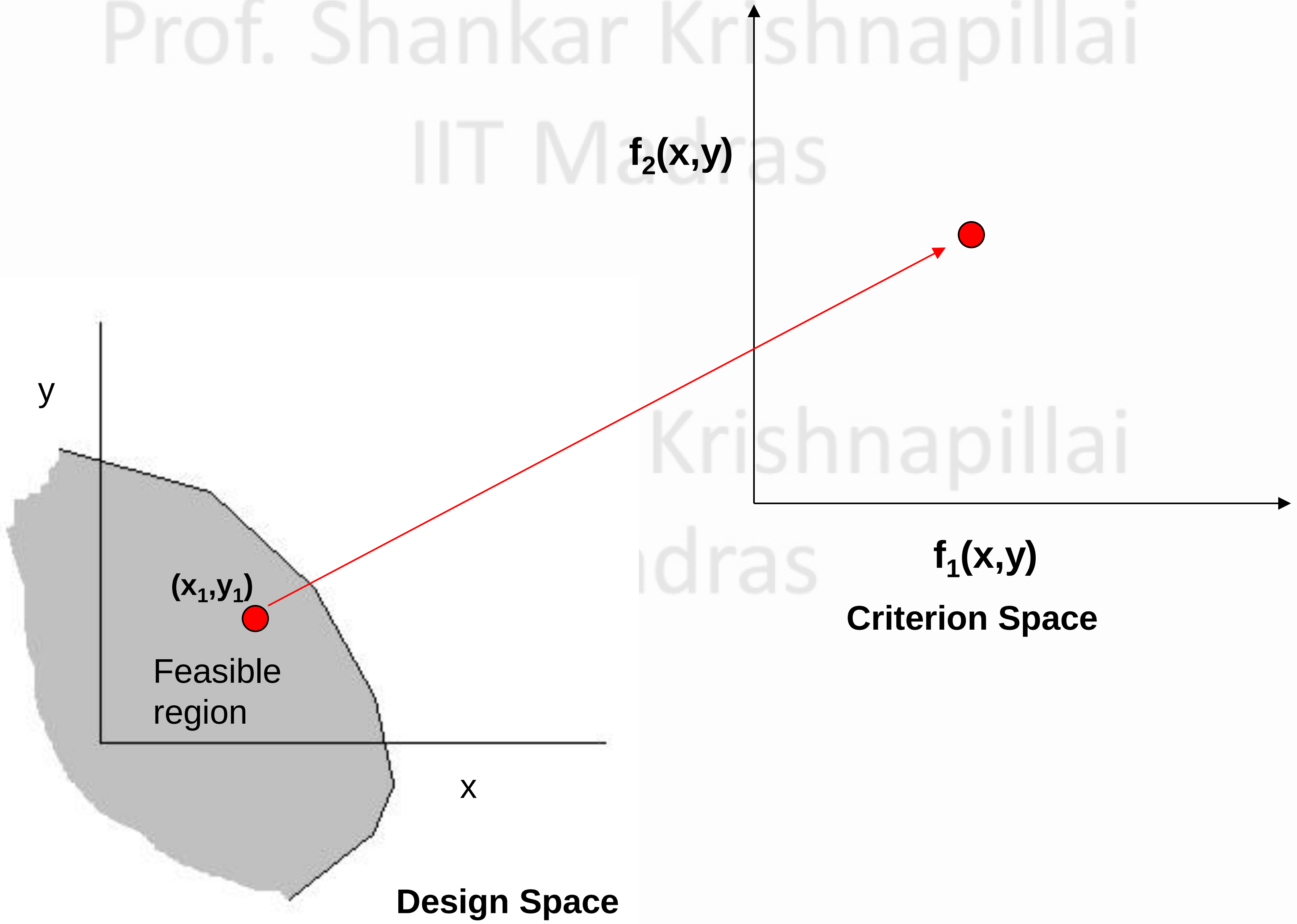


Design Space

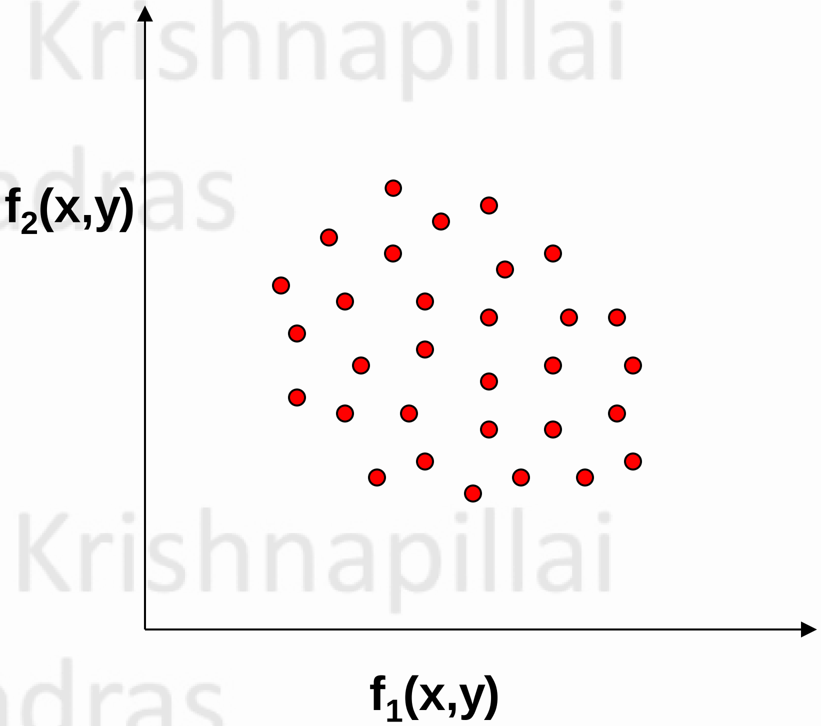


Criterion Space is the "Objective function" space

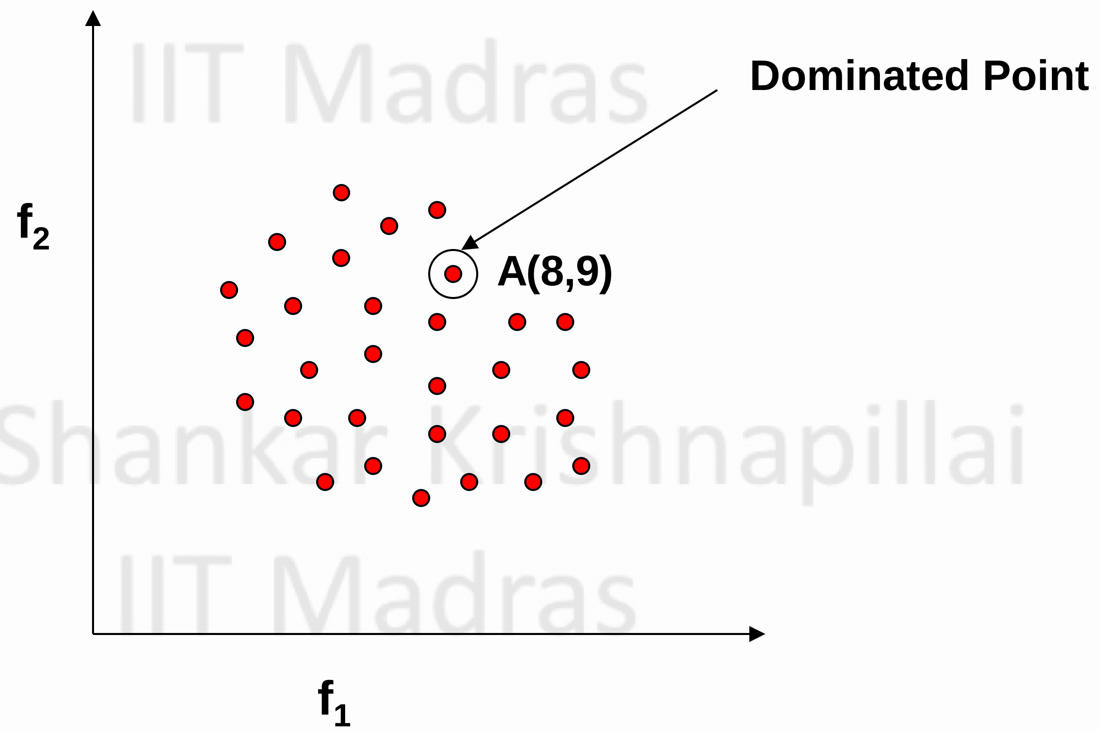
i.e, each (x,y) is converted to f_1 and f_2 by substituting in the expressions for the Objective functions.



Let us assume that every point in the feasible region is mapped to criterion space.

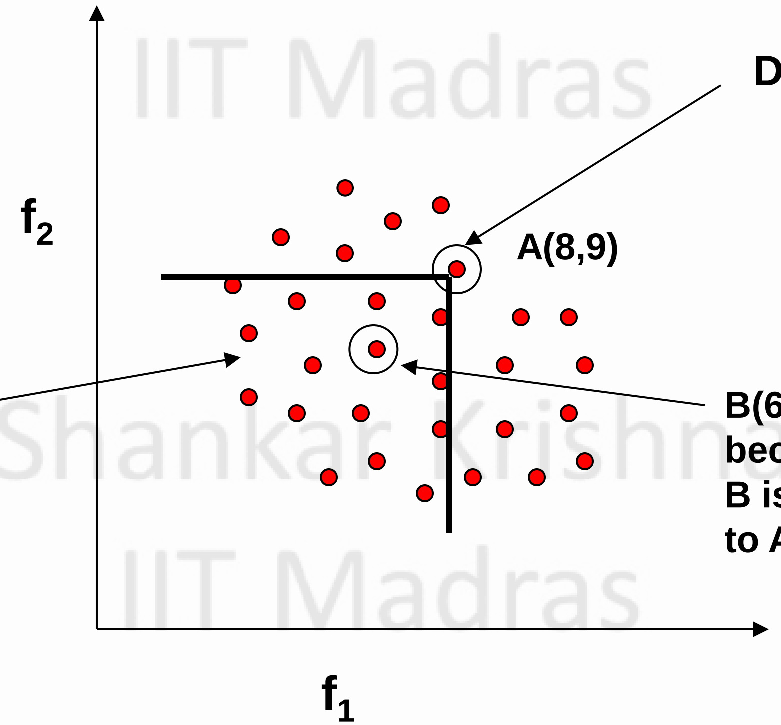


Concept of Dominated and Non-dominated points.



Consider a point $A(f_1,f_2)=(8,9)$ in the Objective function space.

Dominated Point



B(6,7) dominates A(8,9) because 6<8 and 7<9.
B is a better minima compared to A

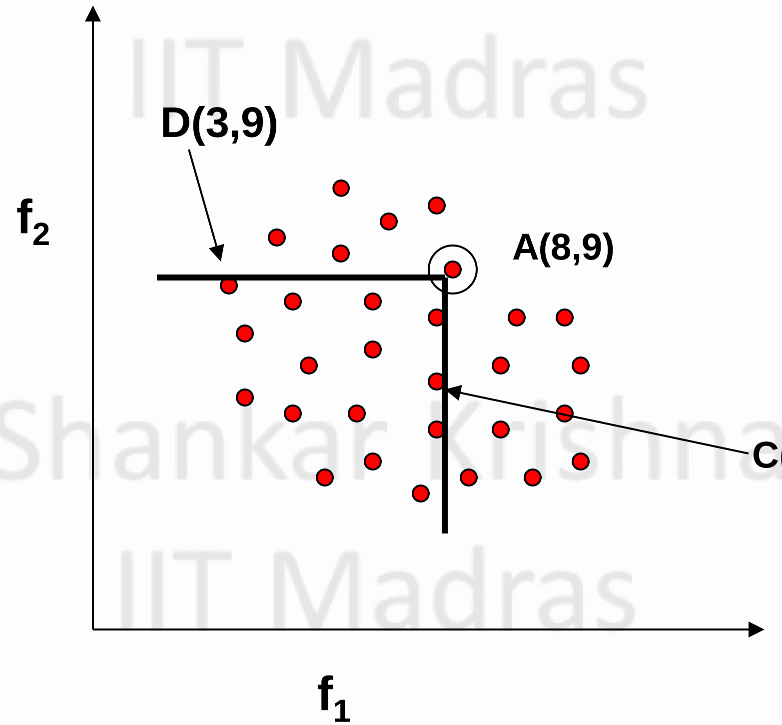
Point A is dominated by all these points in the sector

Prof. Shankar Krishnapillai
IIT Madras

Objective functions

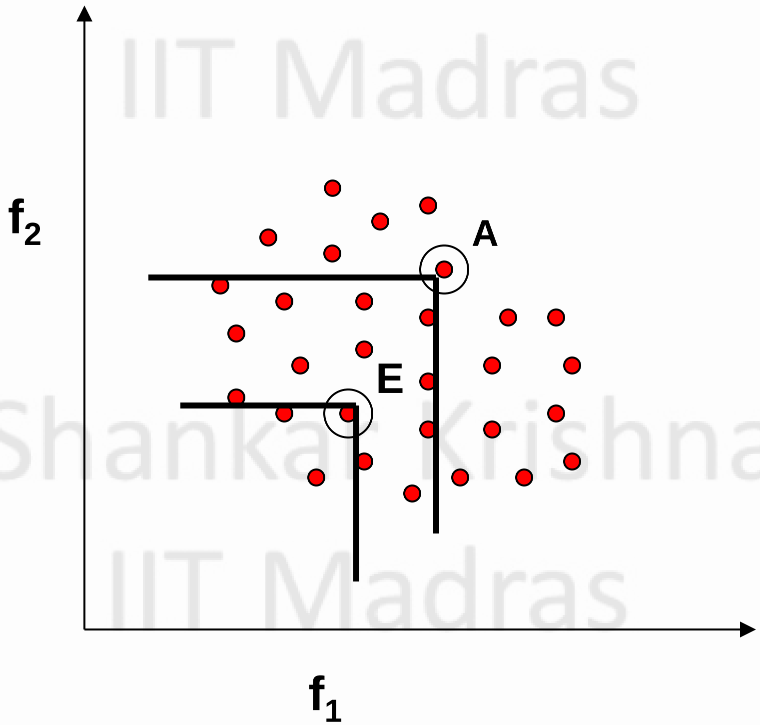
<u>Draw a sector below-left of point A:</u>
Each point in the sector may have f_1 or f_2 (or both) less than A.

Partial Dominance



Objective functions

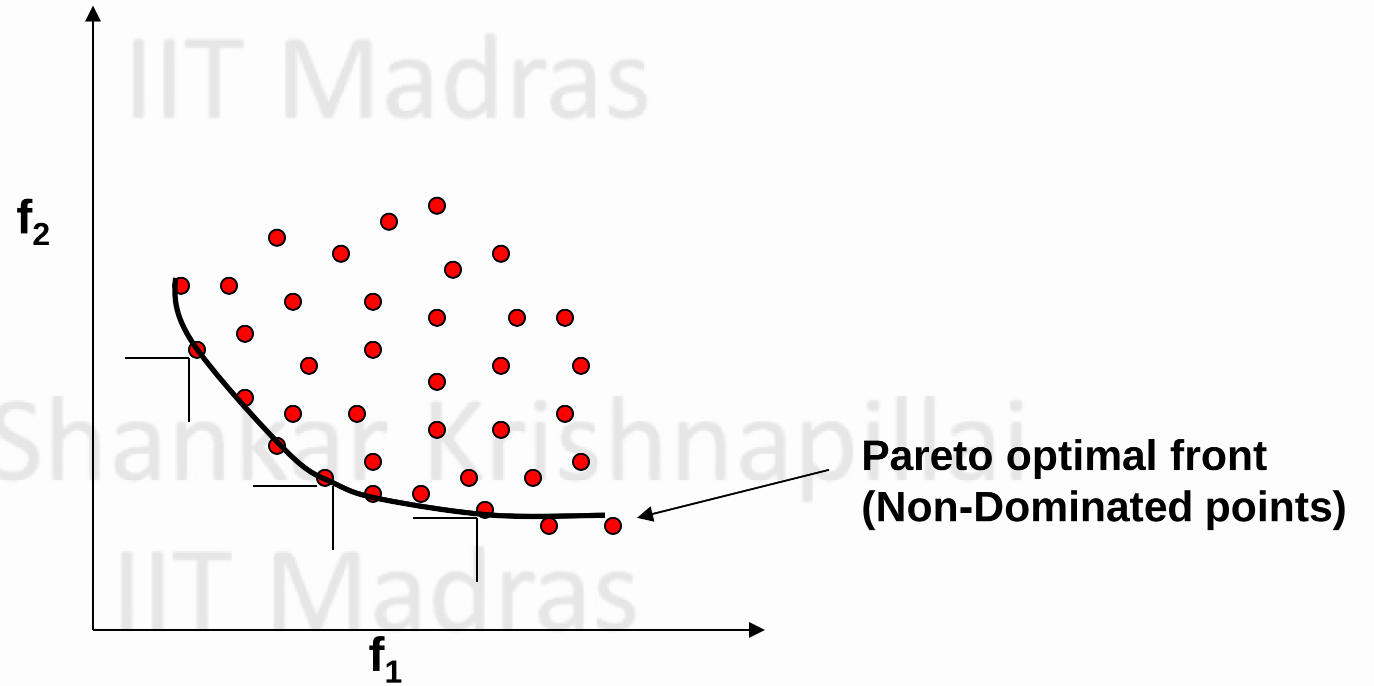
C and D partially dominates A, because their advantage over A is only in less value of one objective function.



Objective functions

We can successively go down to point **E**.
E is dominated by fewer number of points.

Going down, we finally reach the set of **Non-dominated points.**
This is called Pareto Optimal front.
Each point is a unique combination of lowest possible values of f_1 and f_2.
(i.e. each point is not dominated by any other point).



Our goal is to obtain this Pareto Optimal front, by filtering out the Dominated set.

Significant computational effort is usually required for this.

What constitutes the Pareto Optimal front ?
Where does this boundary come from?

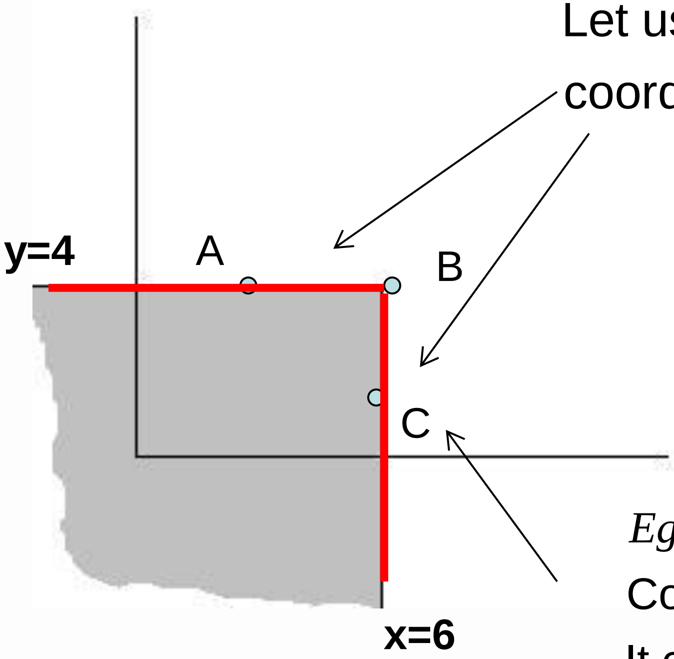
It is a projection of some parts of some Constraint boundary into Criteria Space.
In simple problems we can graphically plot it.

Consider our previous example:

$$\begin{aligned} &\textit{Minimize} \quad f_1 = \sqrt{(x-3)^2 + (y-9)^2} \\ &\qquad\qquad\quad f_2 = \sqrt{(x-10)^2 + (y-1)^2} \end{aligned}$$

Prof. Shankar Krishnapillai
IIT Madras

$$\begin{aligned} &\textit{subject to} \quad g_1 \to x \le 6; \\ &\qquad\qquad\quad\; g_2 \to y \le 4; \end{aligned}$$



Let us convert these constraint boundaries from (x,y) coordinates to (f_1, f_2) coordinates.

Eg.,

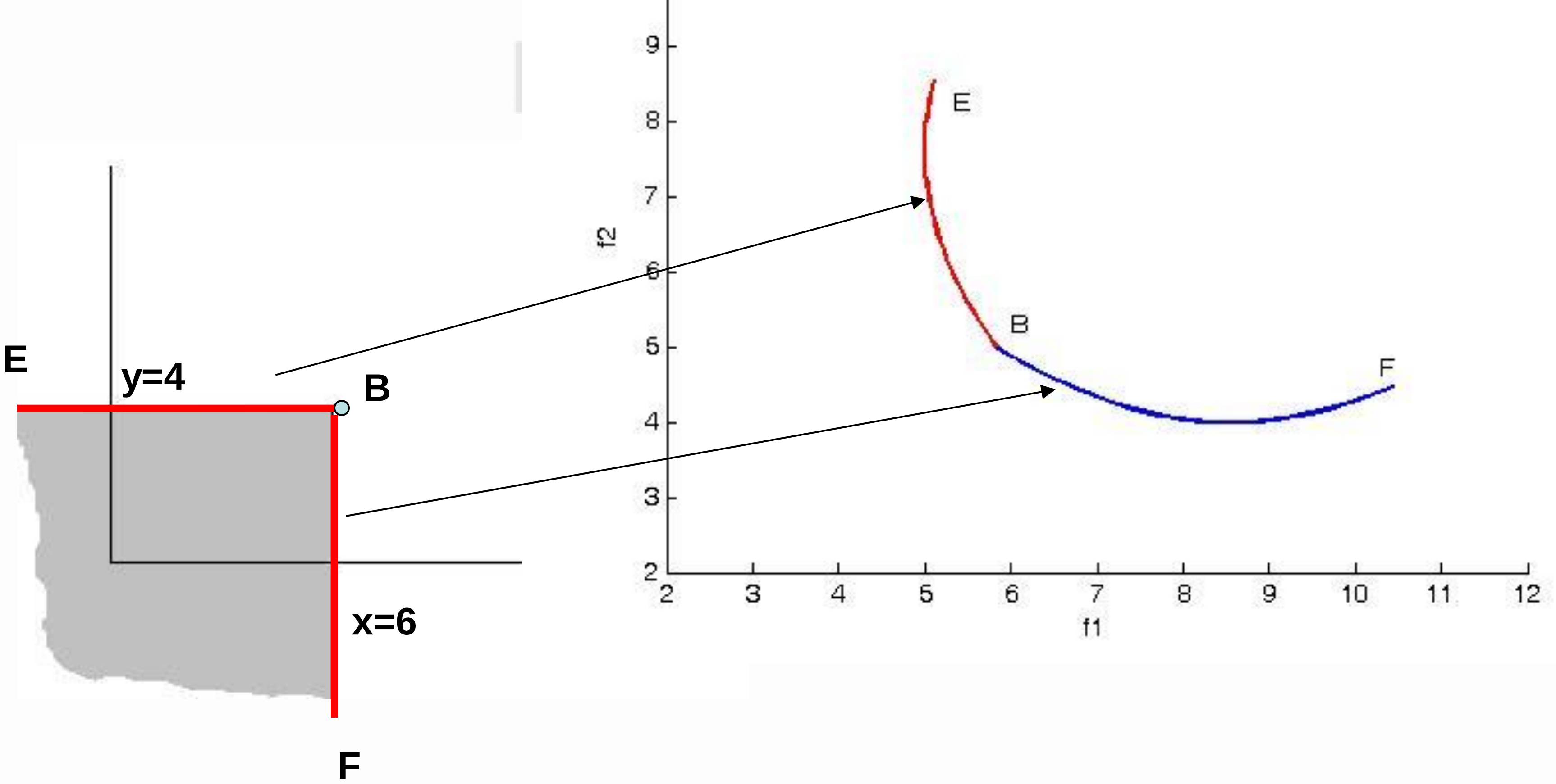
Consider point C(6,1) *i.e*, $x = 6, y = 1$

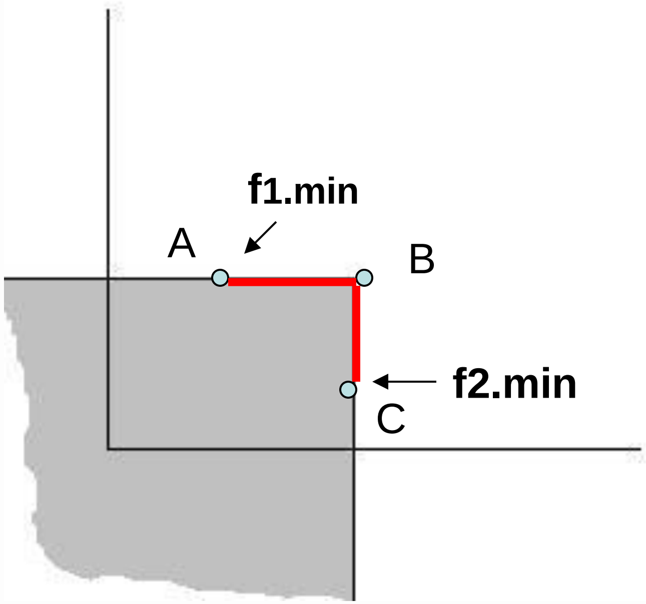
It can be converted to corresponding $(f_1,\ f_2)$ values

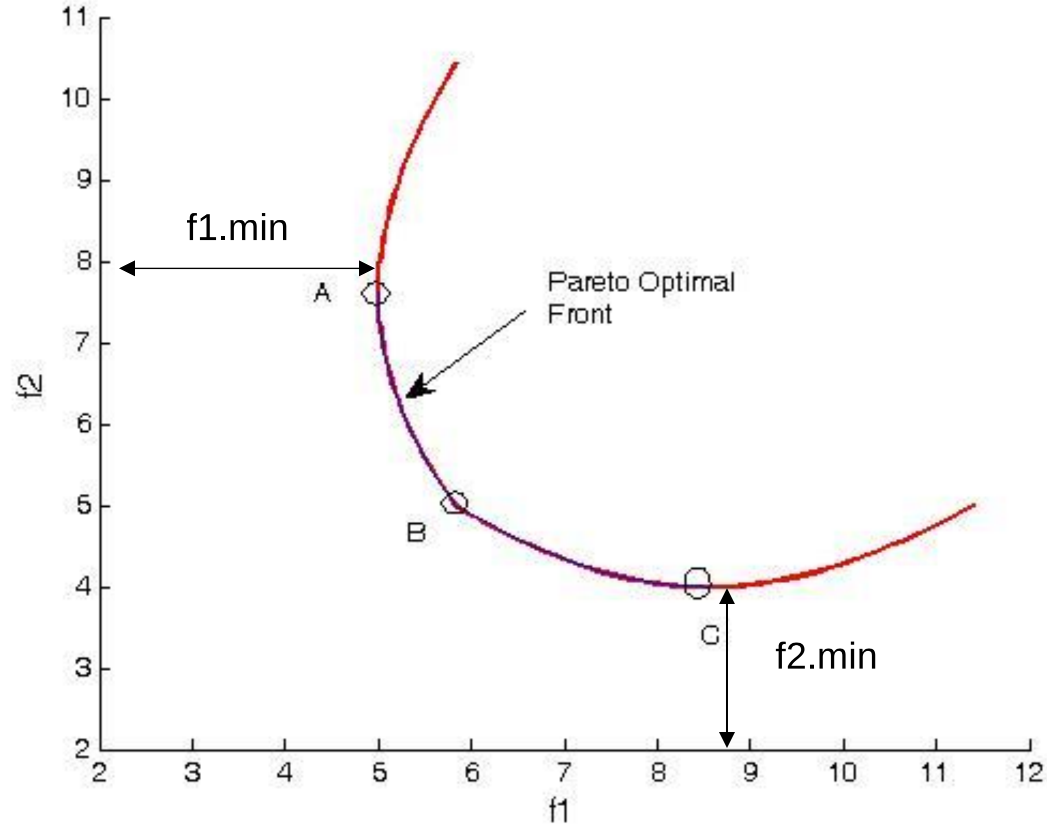
$$f_1 = \sqrt{(6-3)^2 + (1-9)^2} \qquad = 8.54$$

$$f_2 = \sqrt{(6-10)^2 + (1-1)^2} \qquad = 4$$

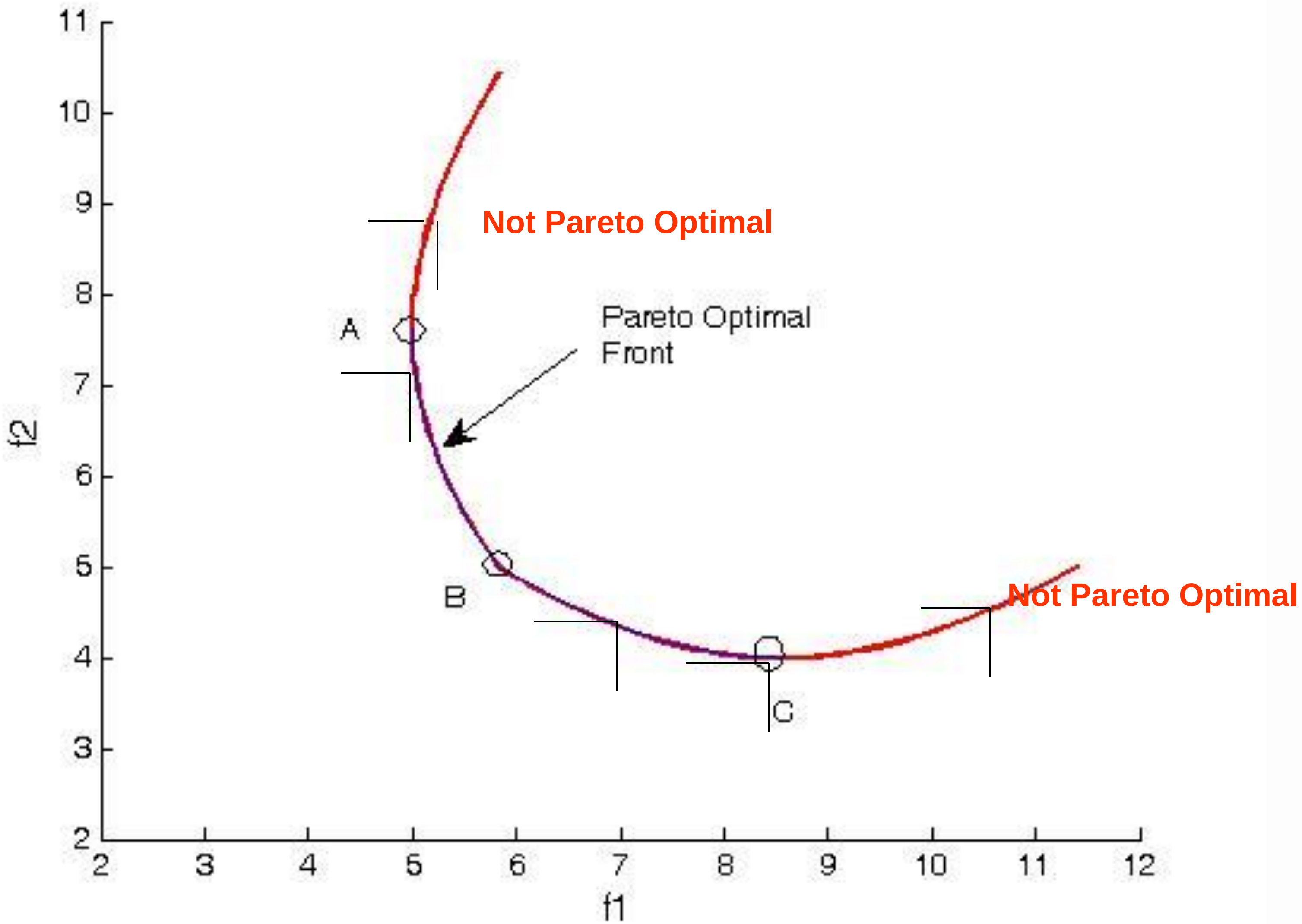
Prof. Sha







The Pareto Optimal part of the curve is from A to C.
'A' is f1.min.
'C' is f2.min.



We can verify which portion of the curve is Pareto optimal by drawing the sectors at each point.

Note: In a simple case like (2 obj. functions, 2 variables) this we can graphically obtain the Pareto front.

If there are many constraints and many design variables, the front has to be calculated using numerical methods.

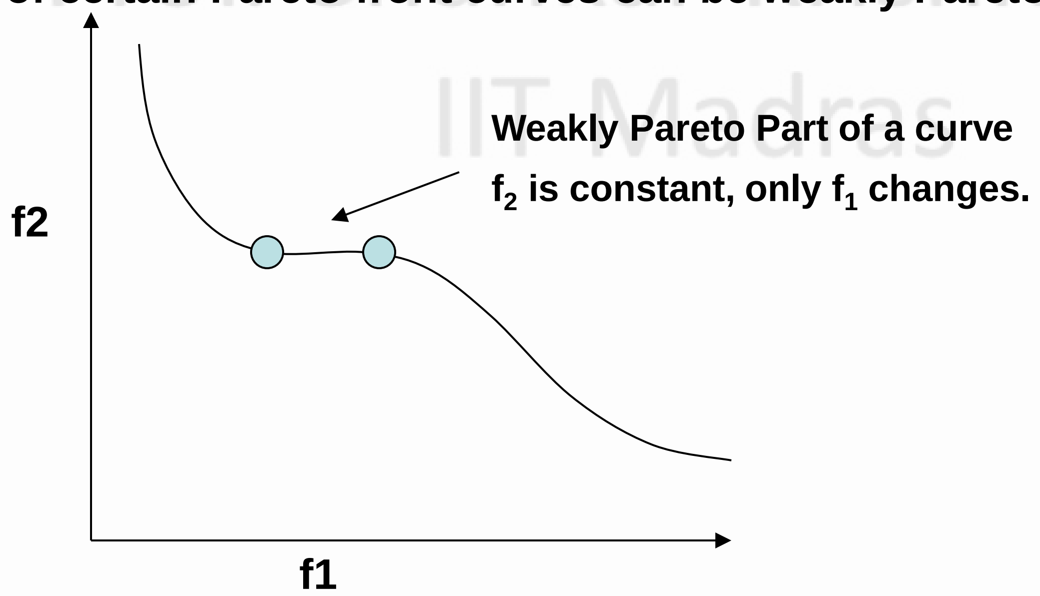
Definition of a Pareto Optimal Point (Strongly Pareto Optimal)

A point $x^{(*)}$ in the feasible space 'S' is said to be Pareto Optimal if there is no other point x in that space which can simultaneously minimize both objective functions better than $x^{(*)}$.

Definition of a Weakly Pareto Optimal Point

A point $x^{(*)}$ is Weakly Pareto Optimal if there is another point x which minimizes one Objective function better than $x^{(*)}$ without any improvement in the other Objective function.

Parts of certain Pareto front curves can be weakly Pareto Optimal.



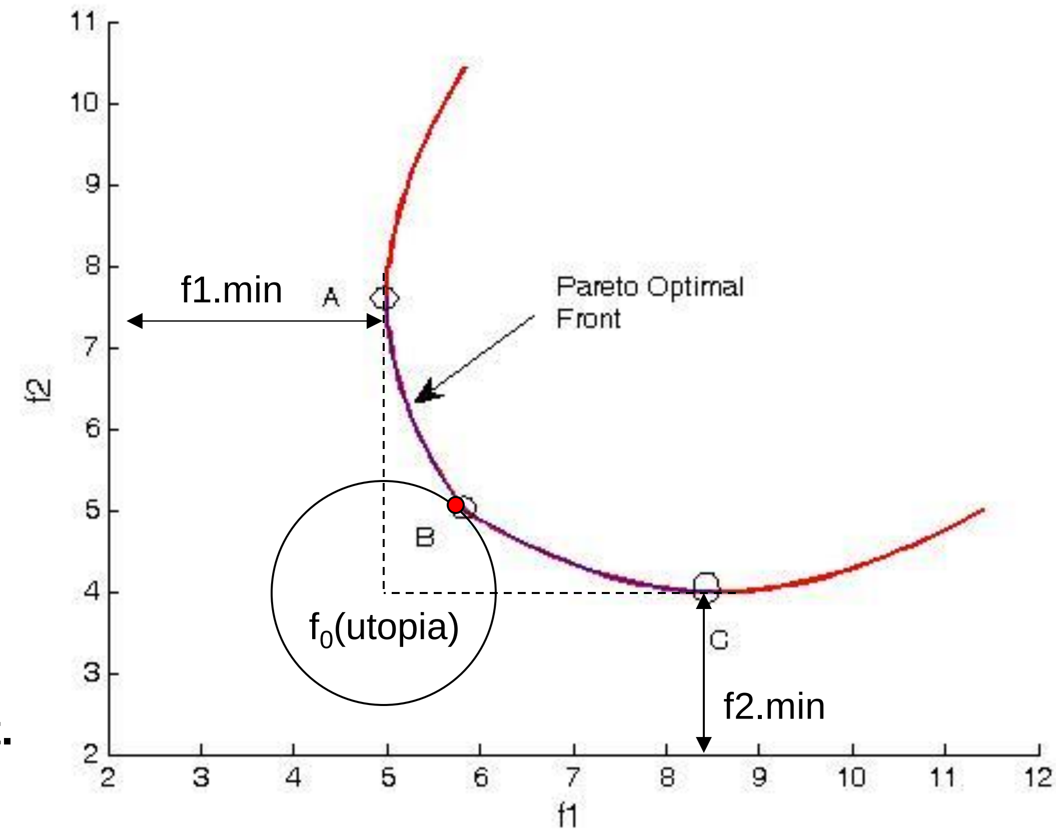
Utopia Point (f_0):

$f_0 = (f_{1.min}, f_{2.min})$.

It represents the lowest minima of both objective functions.

Because objective functions are conflicting, the Pareto front cannot attain the lowest minima of both objective functions.

Hence it will be outside Pareto front.



The point on the Pareto curve having the closest Eucledian distance from Utopia point could be an ideal design choice.

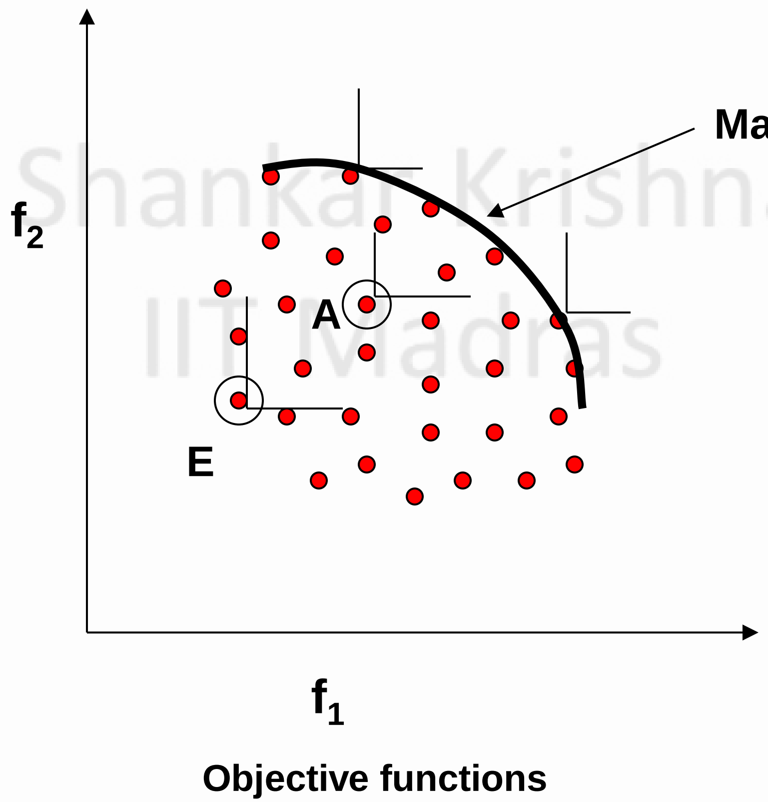
Multi-Objective maximizing problems:

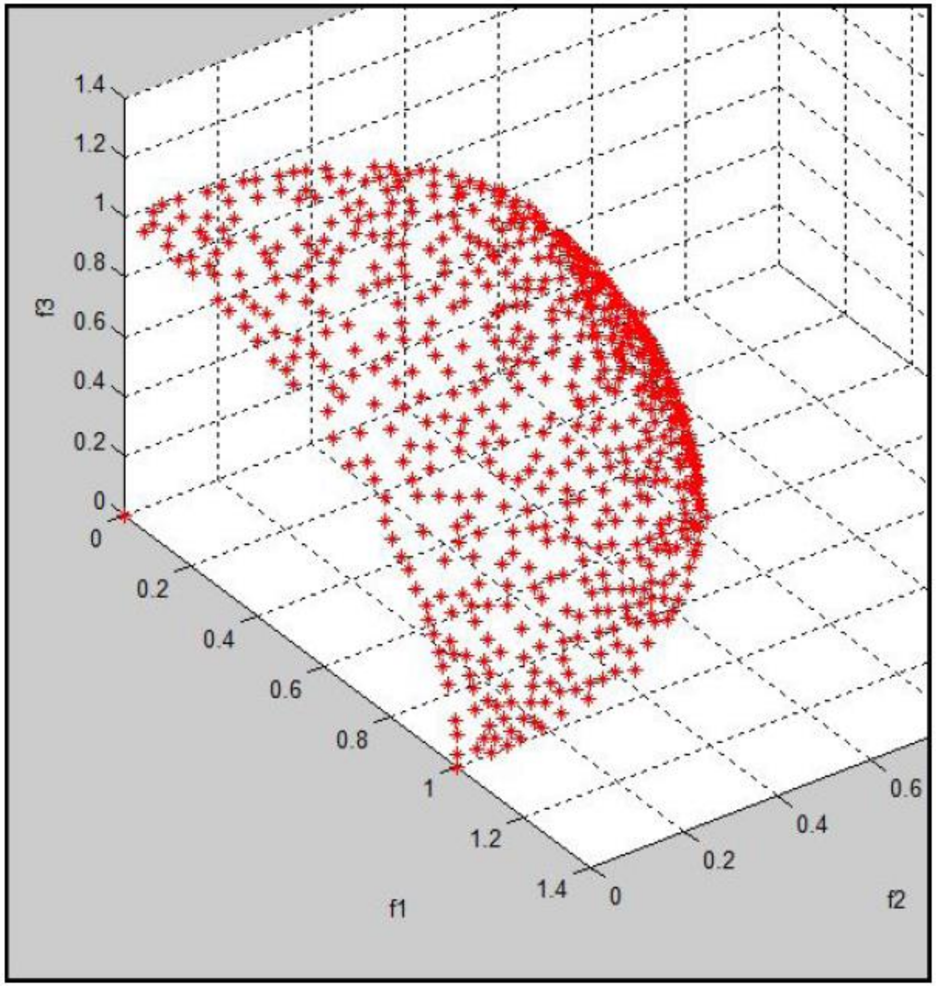
Maximize f_1, f_2

$$
\begin{aligned}
&subject\ \ to:\\
&g_1 \geq 0;\\
&h_1 = 0;
\end{aligned}
$$

The Pareto Optimal front is obtained on the upper right of the Criterion space.

Here A dominates E because it has both f_1, f_2 values higher than E.





Pareto Front in 3 Dimension

Maximizing Pareto front for 3 objective functions
(Ball Bearing Optimization)

Graphical Multi-Objective Optimization

Suitable for:

1. **Simple problems with 2 variables and 2 objective functions.**
2. **Few constraints.**

Method:

1. **Plot feasible region in x-y co-ordinate system.**
2. **Convert the above constraint boundaries to f_1-f_2 co-ordinates.**
3. **Identify the maximizing or minimizing Pareto fronts.**
4. **Identify maximizing and minimizing Utopia points.**

A multi-objective linear programming problem:

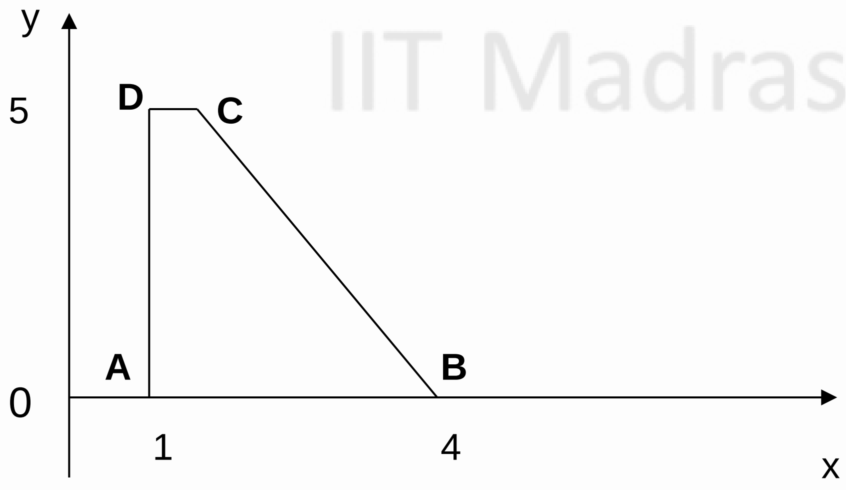
Minimize : $f_1 = 4x - y$

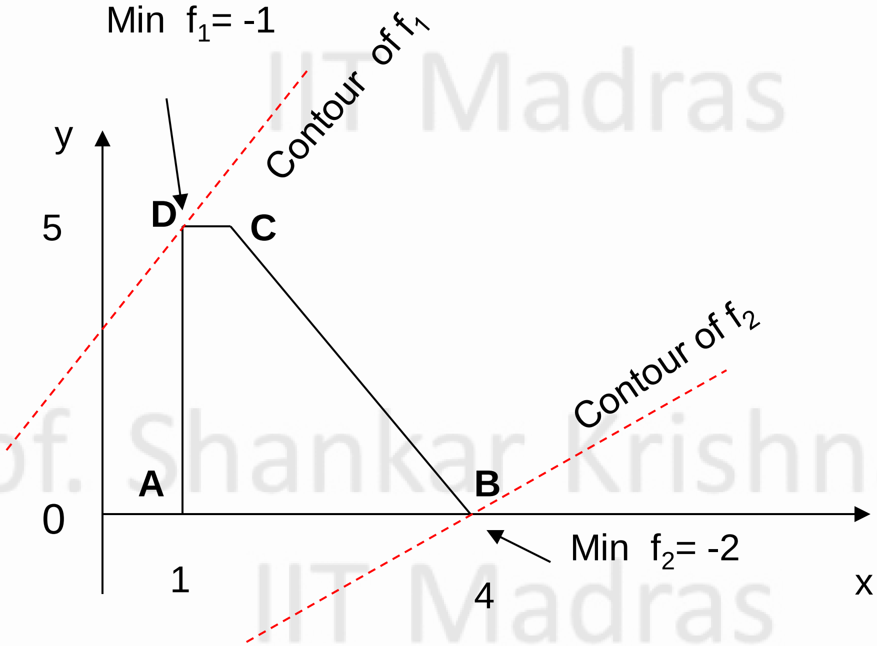
$f_2 = -0.5x + y$

Subject to:

$2x + y \leq 8$; $x \geq 1$; $y \leq 5$; *and* $x, y \geq 0$

First draw the feasible region in the x-y space:





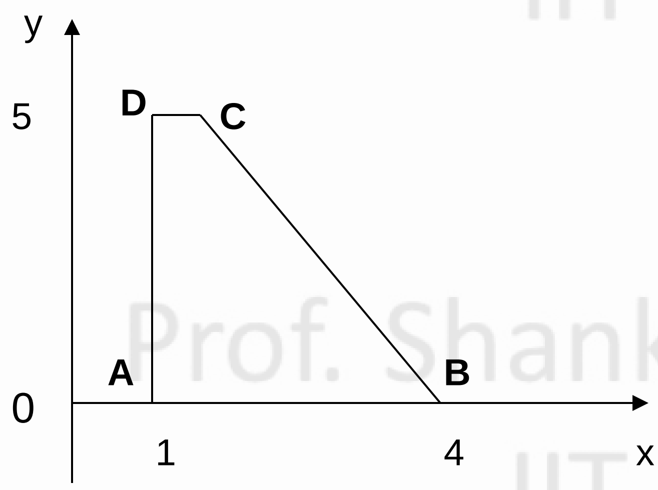
The corners of the feasible region are:

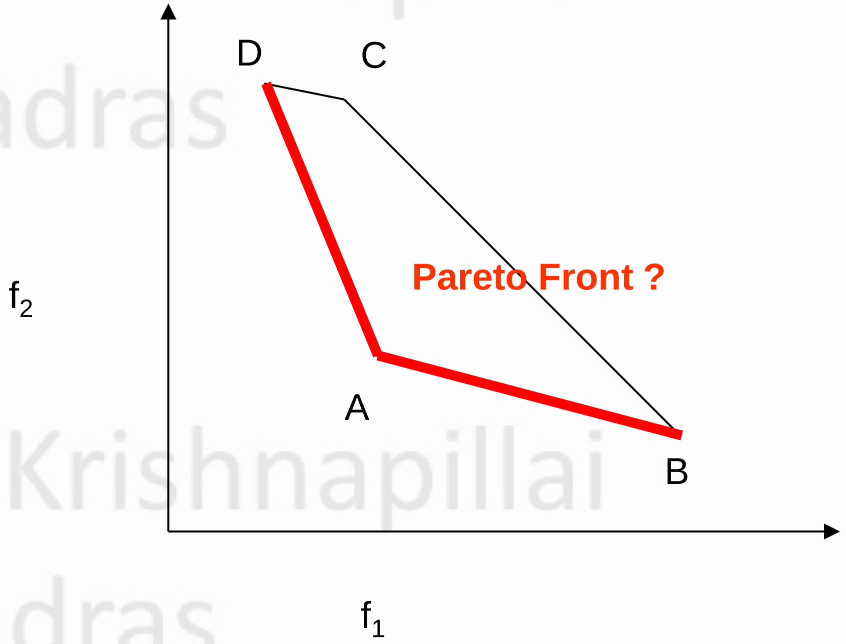
A(1,0)
B(4,0)
C(1.5,5)
D(1,5)

D(1,5) gives $f_{1.min}$ =-1
B(4,0) gives $f_{2.min}$ = -2

How to find the Pareto Front??

Transfer all the A,B,C,D points to f_1, f_2 domain and connect them by lines:

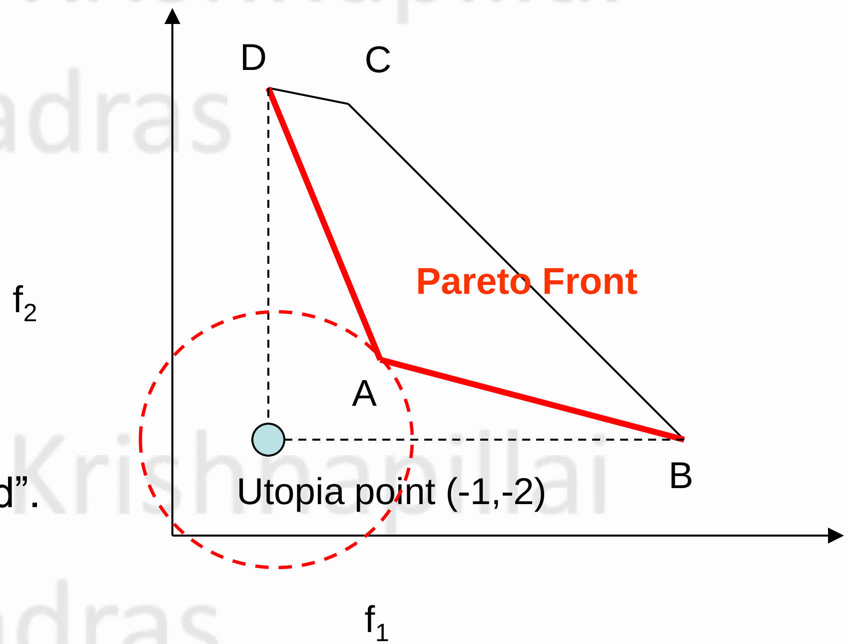




In criteria space:

A(4,-0.5)
B(16,-2)
C(1,4.25)
D(-1, 4)

Prof. Shankar Krishnapillai
IIT Madras



All points on the Pareto front are equally “good”.

D and B correspond to individual minimization of the objective functions.

The point ‘A’ on the Pareto front closest to the Utopia point, is a good choice for the designer.

The (x,y) value corresponding to A i.e., x*=(1,0), is the solution to the problem

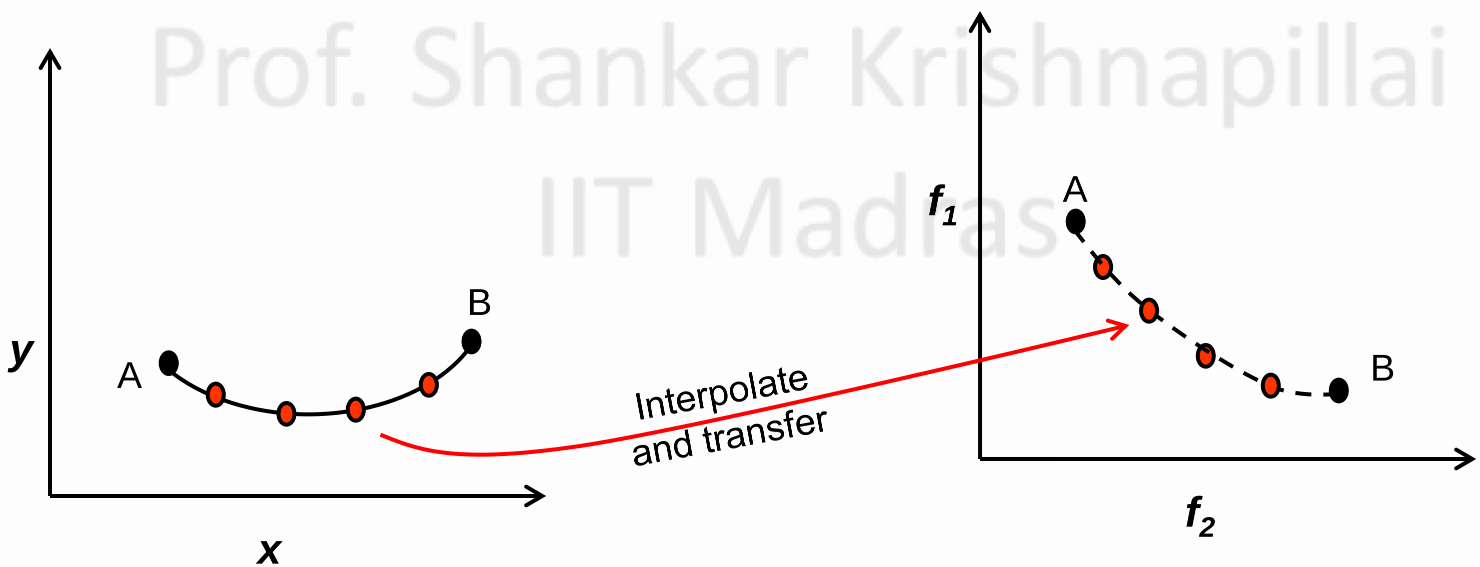
Prof. Shankar Krishnapillai
IIT Madras

Note:

Graphical Multi-objective linear problems are easy, as we only have to transfer the vertices and connect them by lines.

But in case of *non-linear* objectives OR *non-linear* constraints (or both), we cannot do that.

We have interpolate the curve at a few points, and convert them to (f_1, f_2) coordinates and join them with smooth line. This is the general method.



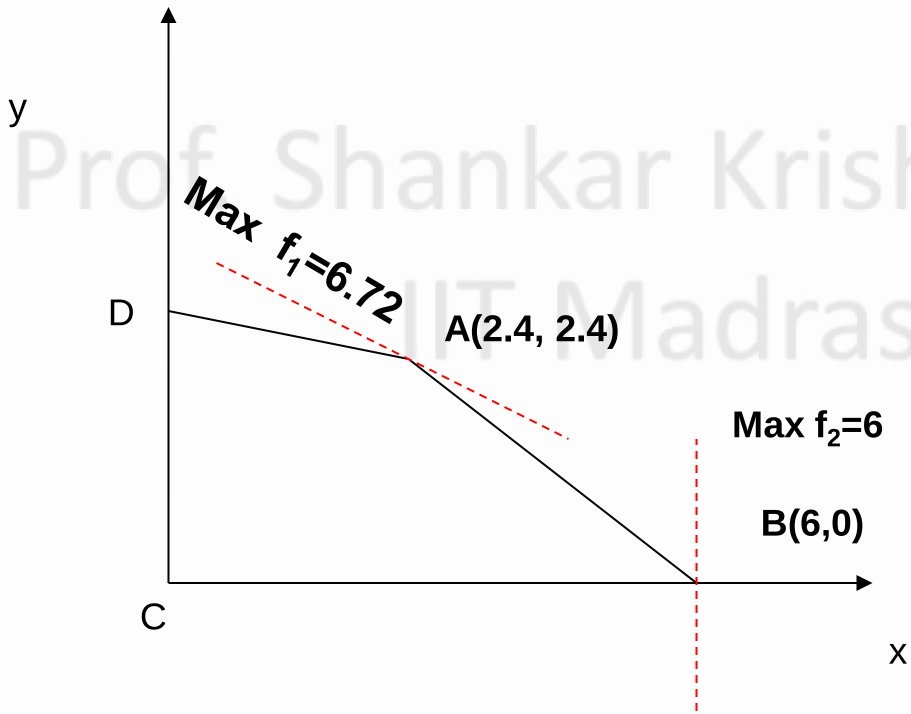
Example of a Maximizing Problem:

Maximize $f_1=0.8x+2y$

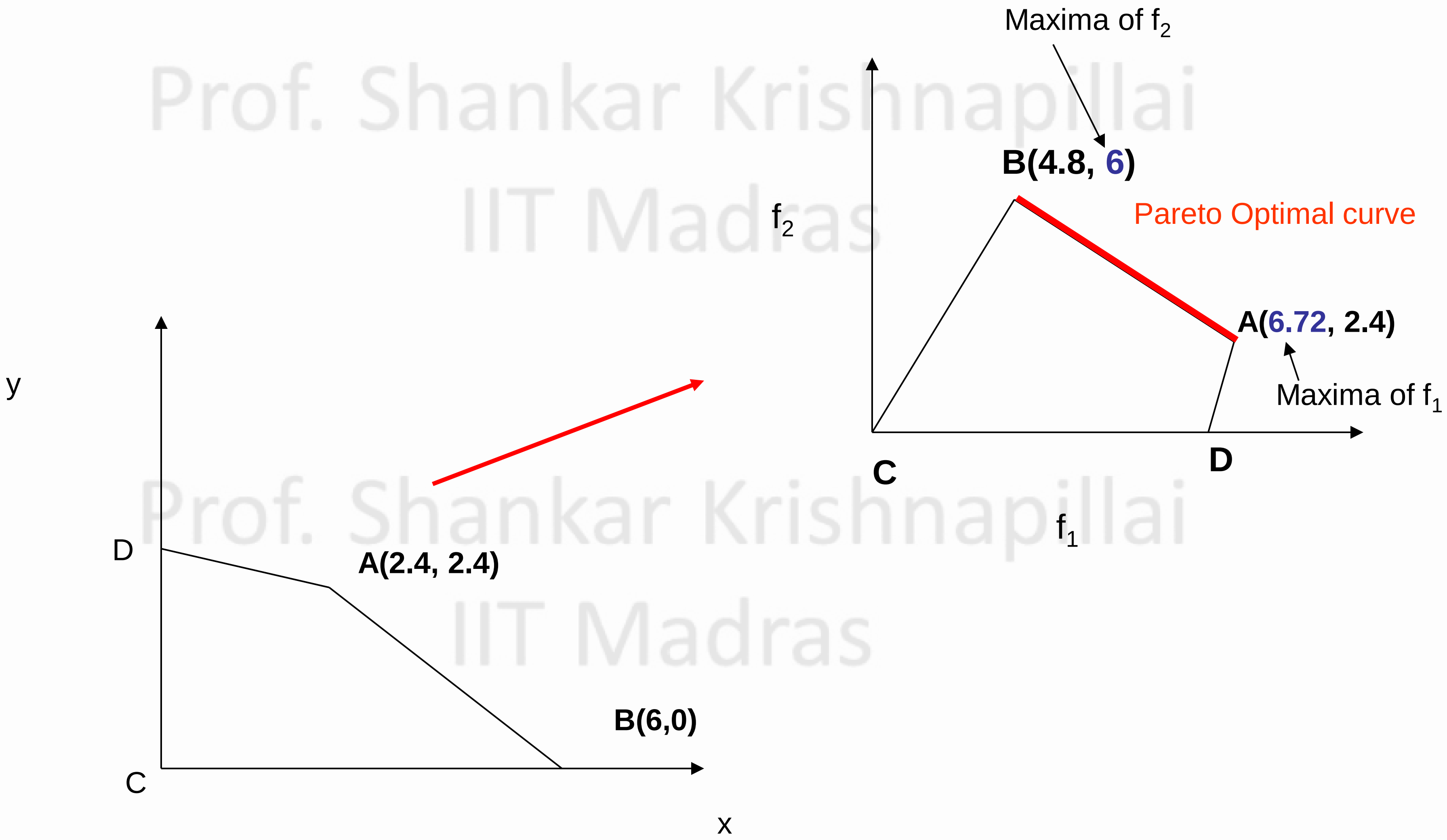
$f_2=x$

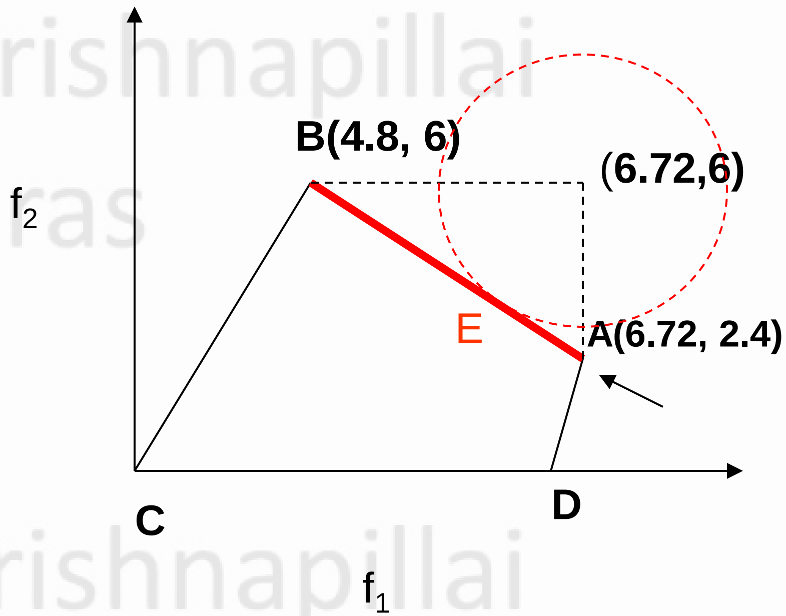
Subject to: $2x+3y \leq 12$

$x+4y \leq 12, \quad (x,y \geq 0);$



Transfer the x-y constraint boundaries to f_1, f_2 domain.





Here, utopia point is ($f_{1.max}$, $f_{2.max}$)

i.e., (6.72,6).

An ideal candidate point would be ‘E’
which is nearest to Utopia point.

The (x,y) cordinates corresponding to E is the solution to this problem.

Weighted Sum method

A common method of Multi-Objective Optimization is by using the Weighted Sum method.

We minimize the weighted sum of the Objective functions:

$$\text{Min} \sum_{i=1}^{k} w_i f_i \quad (k \text{ objective functions})$$

$$\text{and} \sum w_i = 1$$

First, the Objective functions must be of the same 'scale'(i,e comparable range of values).

In such a case, the theorem says that, for *positive values* of weights, for convex feasible regions, the Optimal solution of the weighted problem gives a point on the Pareto Optimal front .

Using different weights, we get different points on the Pareto front.

How can we normalize the Objective functions?

1. Divide with the mean value of the i^{th} Obj. function;

$$f_i^{Norm} = \frac{f_i(x)}{f_{i.mean}} = \frac{f_i(x)}{\left(f_{i.\max} + f_{i.\min}\right)/2}$$

2. or,to normalize in the range 0 to 1,

$$f_i^{Norm} = \frac{f_i(x) - f_{i.\min}}{f_{i.\max} - f_{i.\min}}$$

Prof. Shankar Krishnapillai
IIT Madras

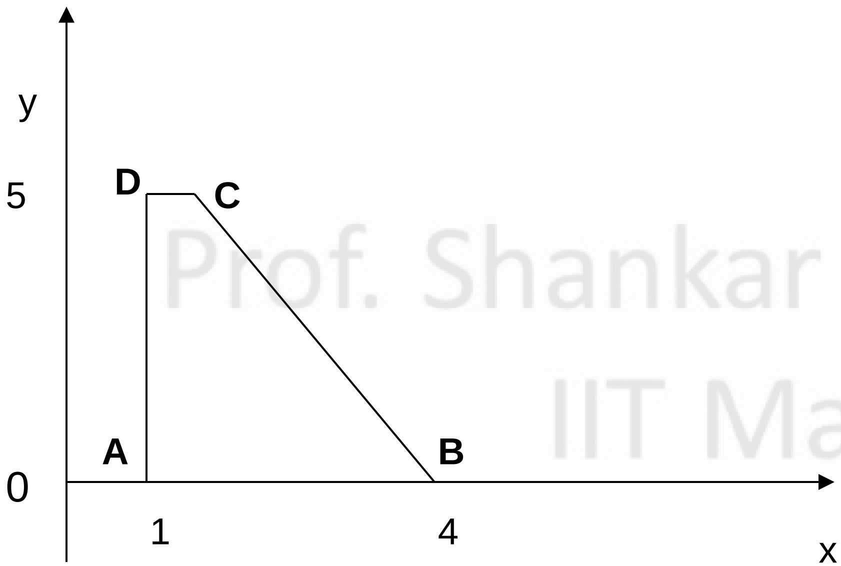
(We have to find the individual maxima and minima of each obj function i
by sampling the domain, or do a seperate optimization for each obj function i.)

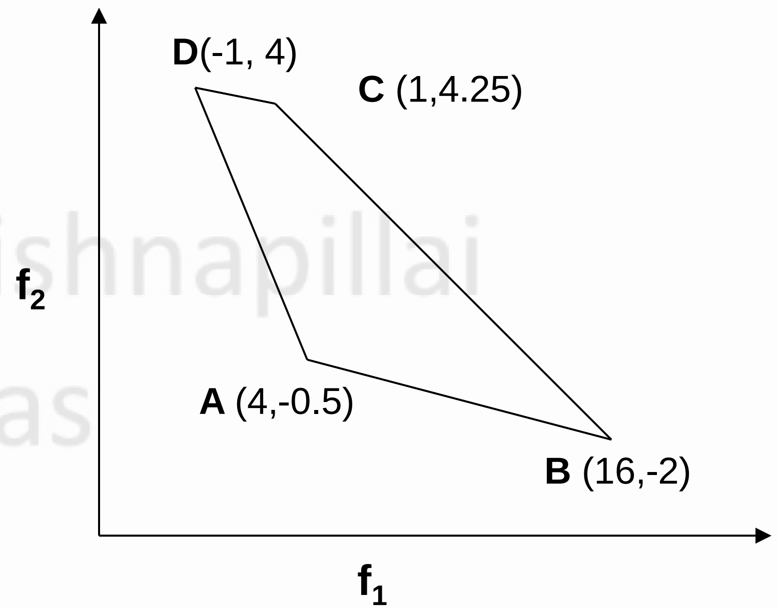
Multi-Objective problem solved with method of weights:

Minimize : $\boxed{\begin{array}{l} f_1 = 4x - y \\ f_2 = -0.5x + y \end{array}}$

Subject to:

$2x + y \le 8; \qquad x \ge 1; \qquad y \le 5; \quad \textit{and} \quad x, y \ge 0$





Note:
We can get same results by individually maximizing or minimizing f_1 and f_2 , using a single objective algorithm.

We get:
$f_{1.min} = -1$, $f_{1.max} = 16$
$f_{2.min} = -2$, $f_{2.max} = 4$

Combined weighted objective function is obtained as follows:

$$Min: \boxed{w_1\, f_1^{Norm} + w_2\, f_2^{Norm}}$$

$$= w_1 \frac{f_1(x,y)}{\left(\dfrac{f_{1.\min} + f_{1.\max}}{2}\right)} + w_2 \frac{f_2(x,y)}{\left(\dfrac{f_{2.\min} + f_{2.\max}}{2}\right)}$$

$$= w_1 \frac{4x - y}{\left(\dfrac{-1+16}{2}\right)} + w_2 \frac{-0.5x + y}{\left(\dfrac{-2+4}{2}\right)}$$

$$= w_1 \frac{4x - y}{(7.5)} + w_2 \frac{-0.5x + y}{(1)}$$

Let us minimize with weights $w_1 = 0.5,\ w_2 = 0.5$

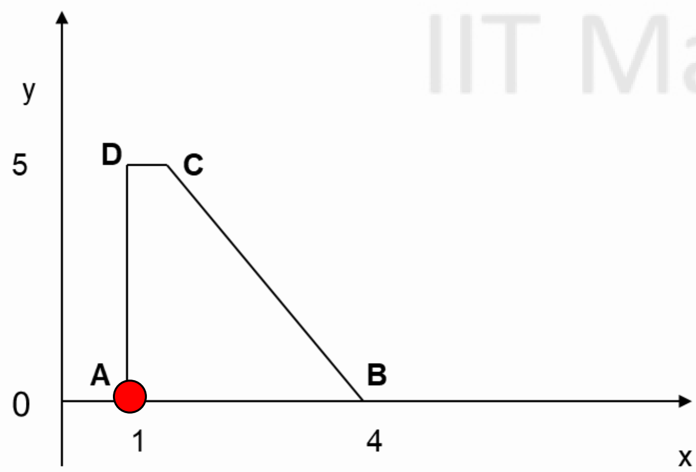
$$Min:\ 0.5 \times \frac{4x - y}{7.5} + 0.5 \times \frac{-0.5x + y}{1}$$

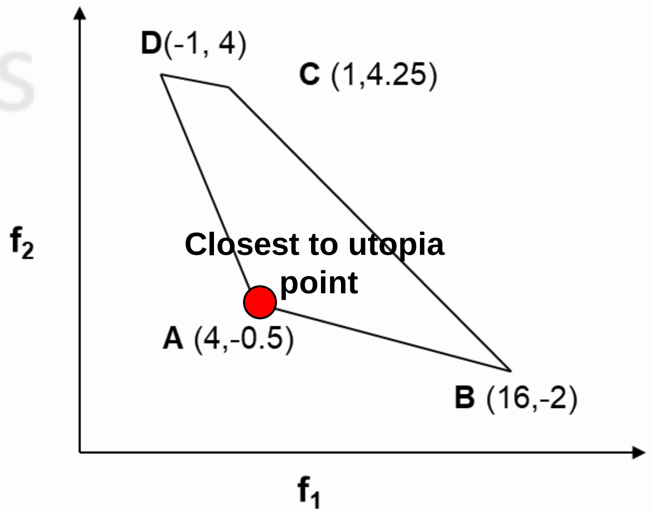
$$:\ \boxed{0.01667x + 0.4333y}$$

$Constraints:$

$$2x + y \le 8; \qquad x \ge 1; \qquad y \le 5; \qquad and \quad x, y \ge 0$$

$Solution\ (x^*, y^*)\ is\ (1,0).\quad In\ (f_1, f_2)\ cords\ it\ is\ closet\ to\ Utopia\ point.$





Goal Programming

It is a practical approach to Multi-objective Optimization.

The Objective functions are given targets (or Goals);
Objective functions then become constraints with the goals as maximum limit they can attain.

The deviations from the Goal are represented by Slack variables which are minimized. If slack variable is zero, goal is attained.

By examining the solved slack variables, we can say wether the Goals are attainable. If not, Goals have to be revised.

With Inequality Constraints

In this case we need to specify the maximum values the Objective function can attain.

Eg. Stress must be not exceed 500Mpa.

Cost of Production must not exceed Rs 1000 per unit.

Example with a Linear Programming Problem:

x and y are the numbers of two different items produced in a factory.

Consider 3 Objective functions.

$f1=7x+3y$ (time of production)

$f2=10x+5y$ (cost of transportation)

$f3=5x+4y$ (machining cost)

constraints:

$(x,y \geq 0)$ (usual LPP constraint)

$100x+60y \leq 600$ (some storage constraint)

Suppose from previous experience we know some reasonable 'Goals' to give the Objective functions.

Such as...

f1 becomes $7x+3y \boxed{\leq 42hrs}$

or, $7x+3y+s_1=42 \qquad (s_1 \geq 0)$

f2 become $10x+5y \boxed{\leq 60Rs}$

or, $10x+5y+s_2=60 \qquad (s_2 \geq 0)$

f_3 becomes $5x+4y \boxed{\leq 40Rs}$

or, $5x+4y+s_3=40 \qquad (s_3 \geq 0)$

Now, the new Set of Constraints are :

$$\left.\begin{array}{r} 7x+3y+s_1 = 42 \\ 10x+5y+s_2 = 60 \\ 5x+4y+s_3 = 40 \\ (s_1,s_2,s_3) \geq 0 \end{array}\right| \textit{New constraints}$$

Plus previous constraints

$10x+60y \leq 600$

$(x,y) \geq 0$

Obj. Function is to: $\underline{\textit{Minimize} \quad s_1 + s_2 + s_3}$

This problem can be solved with Matlab *linprog* subroutine.

The solution to this problem is:

$x=6$, $y=0$;

$s_1=0$;

$s_2=0$;

$\boxed{s_3=10;}$

Note: For the 3rd Objective function.

$$5x+4y+s_3 = 40$$

$5x+4y$ is 30. i.e, the attainable goal is 30.

At this stage we have to decide wether to accept these goals, or to repeat with a new set of goals.

Goal Programming with Equality Constraints:

In some cases we may need to set a Goal exactly equal to some value.

In this case Objective functions become Equality constraints.

Consider our previous example:

We set a exact Goal of 10 to f1 and enforce it with a positive and negative deviation.

f1 become 7x+3y+$\boxed{\eta_1 - \delta_1 = 10;}$ $(\eta_1, \delta_1 \geq 0)$

In the same way we give exact goals to the other two Obj. functions.

f2 becomes 10x+5y+$\eta_2 - \delta_2 = 35;$ $(\eta_2, \delta_2 \geq 0)$

f3 becomes 5x+4y+$\eta_3 - \delta_3 = 15;$ $(\eta_3, \delta_3 \geq 0)$

Note: $\eta_1 - \delta_1$ *is the net deviation from the Goal. It can be positive or negative if Goal is not satisfied.*

This is posed as a Goal Programming Problem:

Minimize : $\eta_1 + \delta_1 + \eta_2 + \delta_2 + \eta_3 + \delta_3$ (suitable for linear programming problem)

$\underline{\textit{Subject to the constraints}:}$

7x+3y+$\eta_1 - \delta_1 = 10$;

10x+5y+$\eta_2 - \delta_2 = 35$;

5x+4y+$\eta_3 - \delta_3 = 15$;

$(\eta_{1,2,3}, \delta_{1,2,3} \geq 0)$

Plus previous constraints

10x+60y $\leq$ 600

(x,y) $\geq$ 0

Solve and check the values of ***η*** and ***δ.***
If they are close to zero, then goals are attained.
If not try different goals.

END