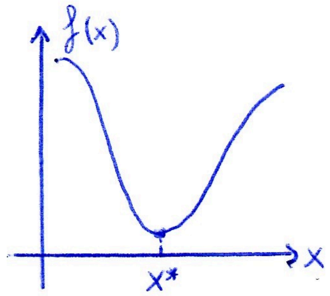
Optimization **Tech**niques

Ecole Centrale Nantes - 2018/19 - Notes by Davide Lanza - Prof. Fouad Bennis

Introduction: Contour Lines, Constraints, etc...



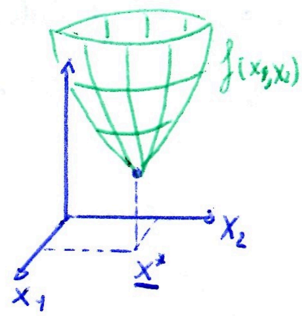
→ How we can find the minimum?

$f'(x^*) = \emptyset \Leftrightarrow x^*$ local minimum

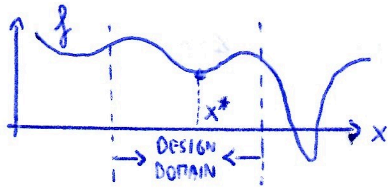
But if we have more dimensions ⤵

2D function:

$$\nabla f(\underline{x}^*) = \underline{\emptyset} = \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} = \begin{pmatrix} \partial f_1 / \partial x_1 \\ \partial f_2 / \partial x_2 \end{pmatrix}$$

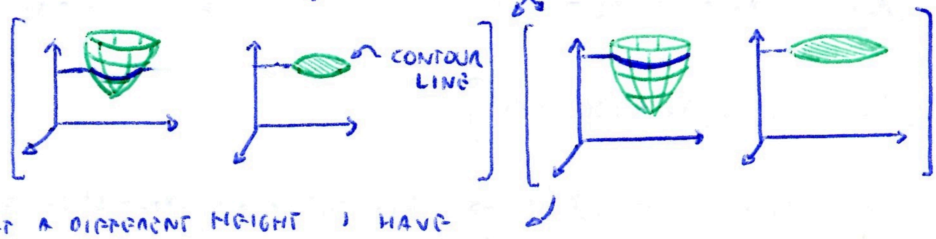


If we have multiple minima? ⤵



→ We normally concentrate only a design domain

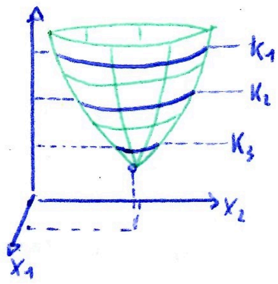
Let's consider the previous $f(x_1, x_2)$ function. If we cut a "slice" of f we'll have ⤵



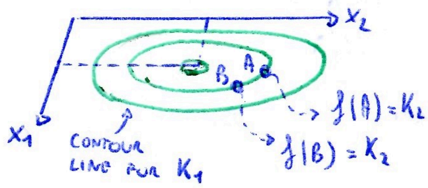
If I cut another slice at a different height I have a different contour line

↳ the contour line is the projection of the function in the contour plane

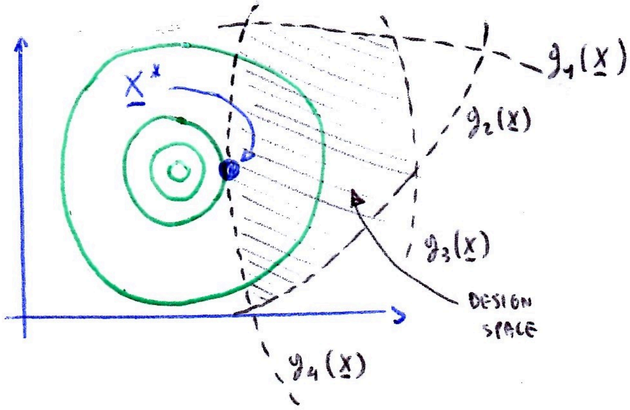
So, we have:



⇒



Normally we consider some constraints that define the design space:



$\underline{X} = (x_1, x_2)$

constraints are function defined as

$$g_i(\underline{X}) \leq \emptyset$$

Here:

$g_4(\underline{X}^*) = \emptyset$ → g_4 is active as the optimus

Example 1: Manufacturing

	Manufacturing time		
	Part 1	Part 2	Max time/week
Machine # 1	10 Parts/min	5 Parts/min	2500
Machine # 2	4 Parts/min	10 Parts/min	2000
Machine # 3	1 Parts/min	1.5 Parts/min	450
Profit/Part	50 €	100 €	

1) Define the objective function (goal function):

$f = \text{Profit}$

2) Define the design variable:

$\bar{X} = (x_1, x_2) = (\text{Part}_1, \text{Part}_2)$ ← number of parts

3) Define the the constraints:

$g_1(\bar{X}) = (10)\, x_1 + (5) x_2 - 2500$ $\quad (g_1 < \emptyset)$

$g_2(\bar{X}) = (4) x_1 + (10) x_2 - 2000$

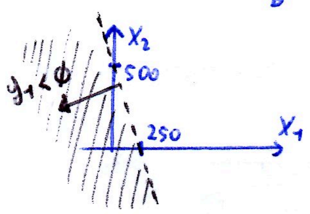
$g_3(\bar{X}) = (1) x_1 + (1.5)\, x_2 - 450$

$g_4(\bar{X}) = -x_1$

$g_5(\bar{X}) = -x_2$

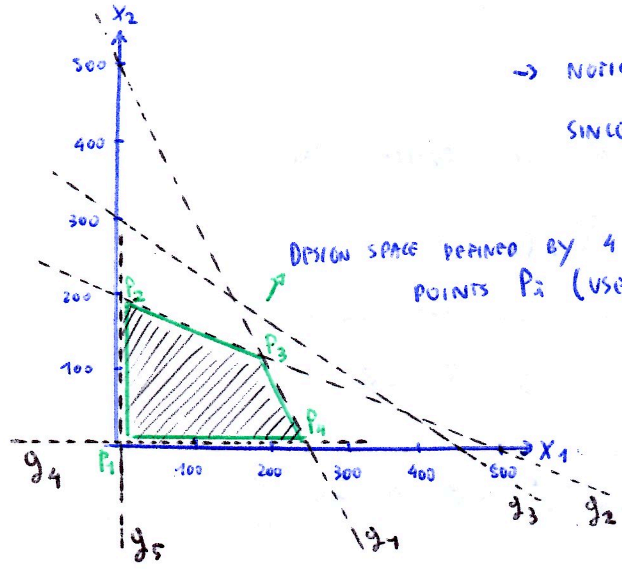
So $\left[f = \text{Profit} = (50)\, x_1 + (100)\, x_2 \right]$

Let's start plotting $g_1(\bar{X})$: → $g_1 = (10)\, x_1 + (5) x_2 - 2500 \Rightarrow \begin{bmatrix} x_1 = \emptyset, & x_2 = 500 \\ x_2 = \emptyset, & x_1 = 250 \end{bmatrix}$



Plotting all together:

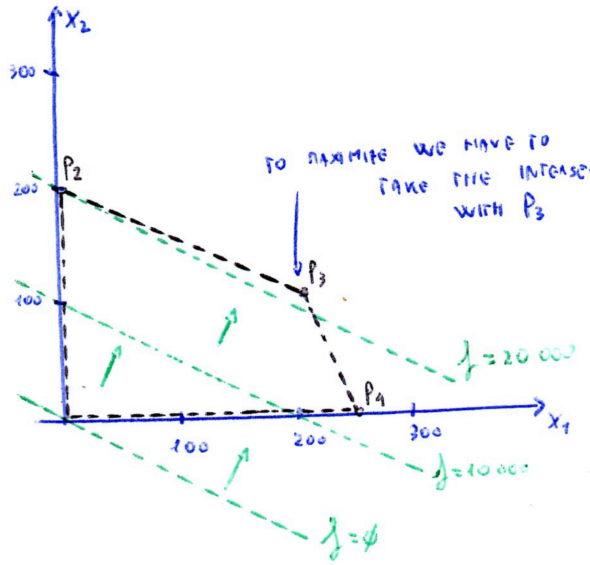
$g_2 \Rightarrow \begin{bmatrix} x_1 = \emptyset, & 200 = x_2 \\ x_2 = \emptyset, & x_1 = 500 \end{bmatrix}$ $\qquad g_3 \Rightarrow \begin{bmatrix} x_1 = \emptyset, & x_2 = 300 \\ x_2 = \emptyset, & x_1 = 450 \end{bmatrix}$



→ Notice how g_3 is not a real constraint, since it does not affect the design space

Design space defined by 4 points P_i (useful later)

AFTER OBTAINING THE DESIGN SPACE LET'S MINIMIZE f:



$$f(\bar{X}) = (50)x_1 + (100)x_2$$

$$f(\bar{X}) = \emptyset \Rightarrow x_2 = -\frac{1}{2}x_1$$

$$f(\bar{X}) = 10 \Rightarrow x_2 = -\frac{1}{2}x_1 + \frac{\cancel{10}}{\cancel{100}}$$

$$f(\bar{X}) = 100 \Rightarrow x_2 = -\frac{1}{2}x_1 + \frac{\cancel{100}}{\cancel{100}}\,1$$

$$f(\bar{X}) = 10000 \Rightarrow x_2 = -\frac{1}{2}x_1 + \frac{100\cancel{00}}{\cancel{100}}$$

etc...

THE OPTIMUM, THANKS TO A MATHEMATICAL THEOREM, IS AT AN INTERSECTION OF g_i AND g_j

↓

WE HAVE 3 POINTS

IN P_3 THE g_1 AND g_2 ARE ACTIVE $(g_1 = \emptyset \wedge g_2 = \emptyset)$

TO OBTAIN "P_3" WE INCREASED THE PROFIT (GEOMETRICALLY) UNTIL IT IS MAXIMIZED ⇒ AT INTERSECTION WITH P_3

WE CAN DO THIS ONLY FOR 2D VARIABLES!

THE SOLUTION, COMPUTED WITH A COMPUTER, IS $\bar{X}^* = \begin{pmatrix} 187.5 \\ 125 \end{pmatrix}$

↓ TO OBTAIN IT ANALYTICALLY

$$\begin{cases} 10\, x_1^* + 5\, x_2^* = 2500 & (g_1) \\ 4\, x_1^* + 10\, x_2^* = 2000 & (g_2) \end{cases} \Rightarrow \bar{X}^* \text{ IS THE SOLUTION}$$

$$f(\bar{X}^*) = 21\,875$$

Example 2: Toys

A COMPANY MANUFACTURER OF TOYS IN WOOD PRODUCES SOLDIERS AND TRAINS:

	TRAINS	SOLDIERS
Selling Price	27 €	21 €
Raw Material	10 €	9 €
General Cost	14 €	10 €
Joinery	1h/Piece	1h/Piece
Finissing	2h/Piece	1h/Piece

MANUFACTURING POWER (hours MAX): JOINERY 80h, FINISSING 100h

Max Soldiers: 40 Soldiers

VARIABLES: x_1 = # TRAINS x_2 = # SOLDIERS

FUNCTION: f = Selling price $= (27)x_1 + (-10)x_1 + (-14)x_1 + (21)x_2 + (-9)x_2 + (-10)x_2 = \;\Rightarrow\; f(\bar{X}) = 3(x_1) + (2)x_2$

CONSTRAINTS:

FINISSING $g_1(\bar{X}) = (2)x_1 + (1)x_2 - 100$

JOINERY $g_2(\bar{X}) = (1)x_1 + (1)x_2 - 80$

Max Soldiers $g_3(\bar{X}) = x_2 - 40$

IS POSITIVE $g_4(\bar{X}) = -x_1$

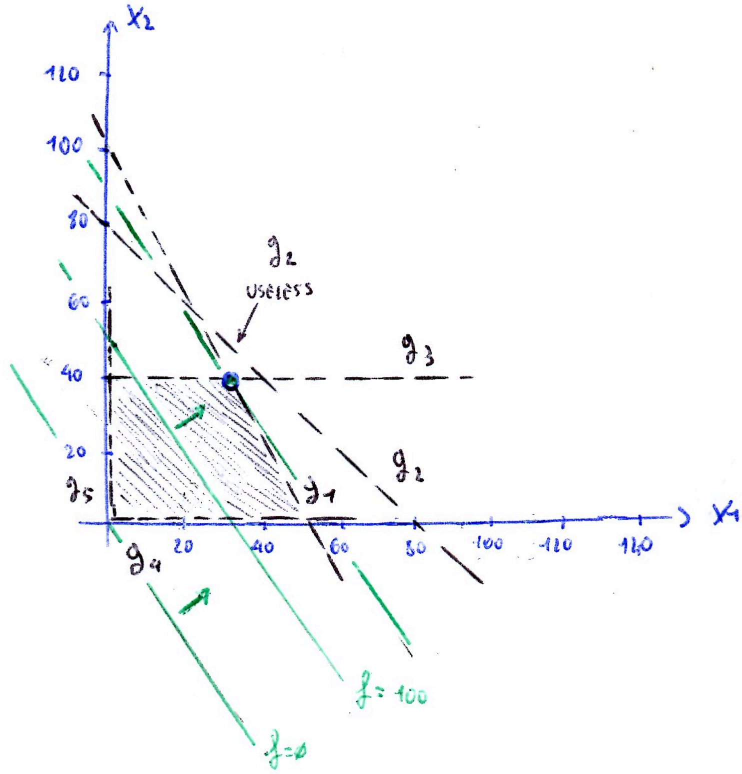
$g_5(\bar{X}) = -x_2$

CONSTRAINT SPACE:

	x_1	x_2
g_1:	$\emptyset / 50$	$100/\emptyset$
g_2:	$\emptyset / 80$	$80/\emptyset$
g_3:		40

$f(\underline{X})$	$x_2(x_1)$
$\emptyset$	$x_2 = -\frac{3}{2}x_1$
10	$x_2 = -\frac{3}{2}x_1 + \frac{10}{2}$ (5)
100	$x_2 = -\frac{3}{2}x_1 + 50$
1000	$x_2 = -\frac{3}{2}x_1 + 500$

etc...

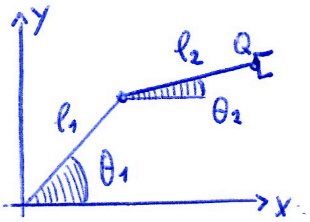


→ g_1 ∧ g_3 ARE THE ACTIVE CONSTRAINTS

$$\begin{cases} 2x_1^* + x_2^* = 100 \\ x_2^* = 40 \end{cases} \rightarrow \cancel{2}x_1^* = \cancel{60}^{30} \rightarrow \underline{X}^* = (30, 40) \qquad f(\underline{X}^*) = 170\text{ €}$$

Example 3: RR Robot

Find the joint coordinates $\theta = (\theta_1, \theta_2)$ s.t. we have $P(x_P, y_P) = (2,3)$



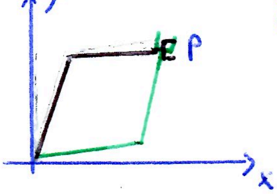
VARIABLES: $\underline{X} = (\theta_1, \theta_2)$

FUNCTION: $f = \| \vec{PQ} \|$

CONSTRAINTS: Joint limits or obstacles (not considered here)

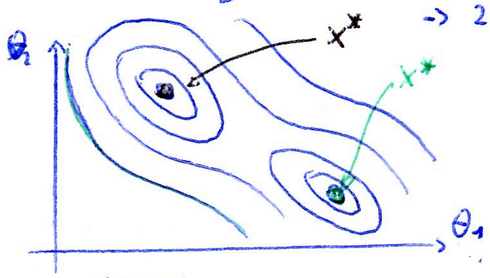
$$Q = \begin{pmatrix} l_1 \cos\theta_1 + l_2 \cos\theta_2 \\ l_1 \sin\theta_1 + l_2 \sin\theta_2 \end{pmatrix} \longrightarrow f = \sqrt{(x_Q - x_P)^2 + (y_Q - y_P)^2}$$

we will find 2 solutions



"ELBOW-UP ELBOW-DOWN" CONFIGS

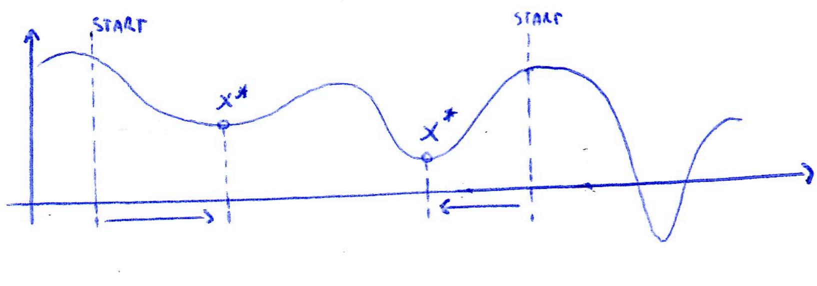
IN FACT WE HAVE →



→ 2 LOCAL MINIMA

AND THERE IS NO GUARANTEE TO FIND THE GENERAL MINIMUM!

∄ ALGORITM THAT GIVES YOU ALL THE MINIMA, ONLY THE NEAREST TO THE INITIAL CONDITION! →



Example 4: Antennas

Find the "best" location of an antenna allowing the connection of 4 customers. Priority for the "best" customer.

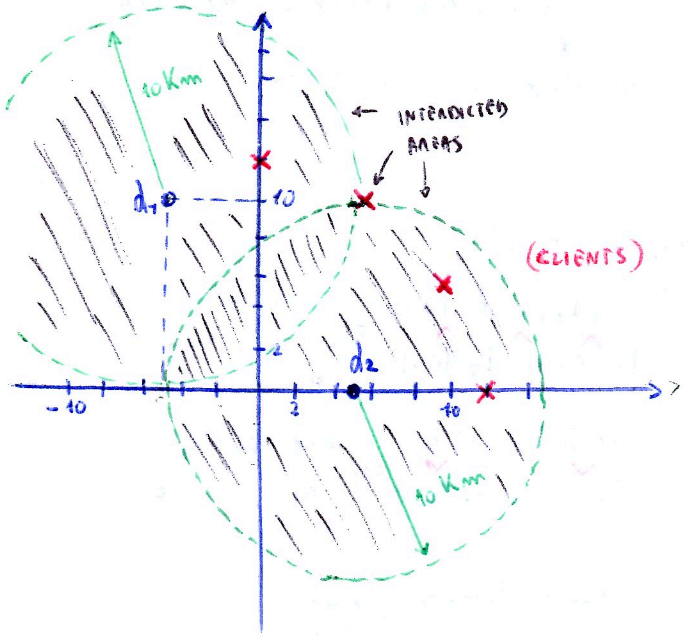
Customer	Coords	Consumption h
1	(5,10)	200
2	(10,5)	150
3	(∅,12)	200
4	(12,∅)	300

Existing antennas at (-5,10) and (5,∅)

Interdiction of place antennas less than 10 Km from the existing ones

Variables: $(x_1, x_2) = (x, y)$ coord of new antenna

Constraints:



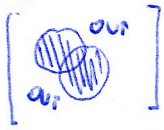
$$g_1(\underline{X}) = 10 - \sqrt{(x_1 - (-5))^2 + (x_2 - (10))^2}$$

antenna d_1

$$g_2(\underline{X}) = 10 - \sqrt{(x_1 - (5))^2 + (x_2)^2}$$

antenna d_2

! is different w.r.t. the previous examples' constraints 'cause here the design space is out, not in →



Function: we have different ways to build f. Starting from:

$$d_i \triangleq (\text{distance customer } i - \text{antenna}) = \sqrt{(x_1 - x_i)^2 + (x_2 - y_i)^2}$$

$$\cancel{f = \sum_{i=1}^{4} d_i}$$ → not good → does not take in account the "best" customers

f weighted on consumption c_i of customer i:

$$f = \sum_{i=1}^{4} c_i d_i$$

OR

$$f = \sum_{i=1}^{4} \frac{c_i}{c_{tot}} d_i \quad \text{with } c_{tot} = \sum_{i=1}^{4} c_i$$

But we have to minimize here d_1, d_2, d_3, d_4 !

it is a multi-objective problem

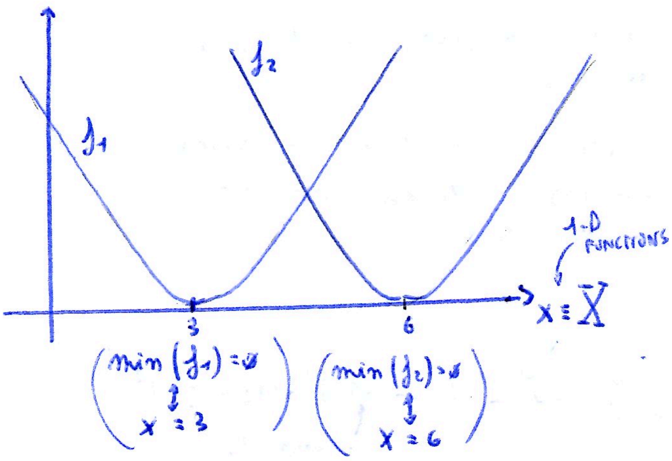
↓

see next section

Introduction to Multi-Objective Optimization

Let's consider $f_1(\bar{X}) = (x-3)^2$ and $f_2(\bar{X}) = (x-6)^2$

We want to find $\bar{X}^*$ minimum of $f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$



- If we choose:

$$F = f_1 + f_2 = 2x^2 - 18x + 45$$
$$\downarrow$$
$$F' = 4x - 18 \Rightarrow x^* = 4.5$$

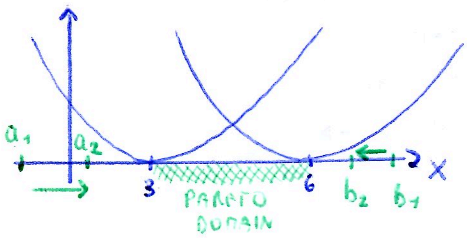
- If we choose:

$$F = f_1 \cdot f_2 = (x-3)^2(x-6)^2 =$$
$$= x^4 - 18x^3 + 117x^2 - 324x + 324$$
$$\downarrow$$

Two $F=\emptyset$ minima in (3) and (6)

None of the two $F(X)$ is appropriate!

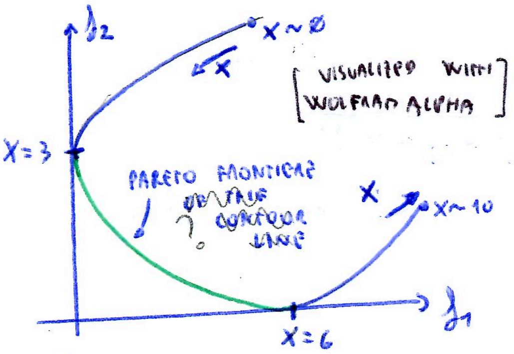
Let's analyse the behaviour of the two minima of f_i:



For $x < 3$: $\begin{cases} f_1(a_2) < f_1(a_1) \\ f_2(a_2) < f_2(a_1) \end{cases} \Rightarrow$ $x=3$ dominates all $x<3$

For $x > 6$: $\begin{cases} f_1(b_2) < f_1(b_1) \\ f_2(b_2) < f_2(b_1) \end{cases} \Rightarrow$ $x=6$ dominates all $x>6$

For $3 < x < 6$: Pareto Domain, where the negotiation begins

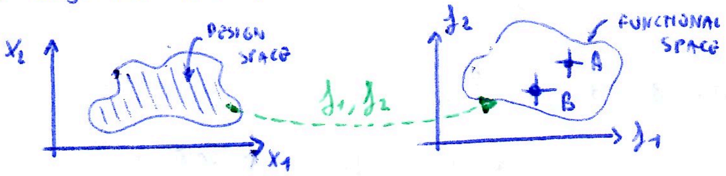


In $x=\emptyset$ $\begin{pmatrix} f_1 = 9 \\ f_2 = 36 \end{pmatrix}$

In $x=10$ $\begin{pmatrix} f_1 = 49 \\ f_2 = 16 \end{pmatrix}$

$\downarrow$

For multi-objectives problem is useful the functional space:



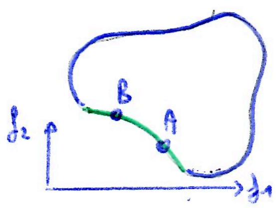
In functional space we say:

If $\begin{rmatrix} \end{rmatrix}$ $\left.\begin{matrix} f_1(B) < f_1(A) \\ f_2(B) < f_2(A) \end{matrix}\right\}$ then B dominates A

$\downarrow$

For those points is easy to compare

But for two points in the Pareto frontiere is difficult:



$\Rightarrow \begin{matrix} f_1(B) < f_1(A) \\ f_2(B) > f_2(A) \end{matrix}$!

Scalarizing Multi-Objective Functions

In order to find $\min(f_1, f_2)$:

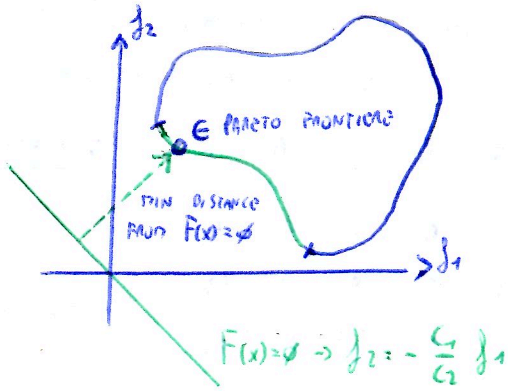
a) **Linear scalarization:** $F(\bar{X}) = c_1 \cdot f_1 + c_2 \cdot f_2$

↳ you can change c_1 and c_2 and you'll find other solutions

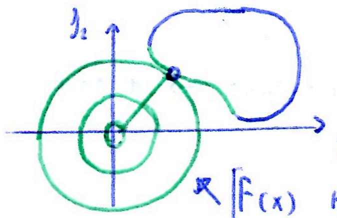
BUT

all solutions $\in$ Pareto frontiere

? I made up this term



b) **Quadratic scalarization:** $F(\bar{X}) = c_1 \cdot f_1^2 + c_2 f_2^2$



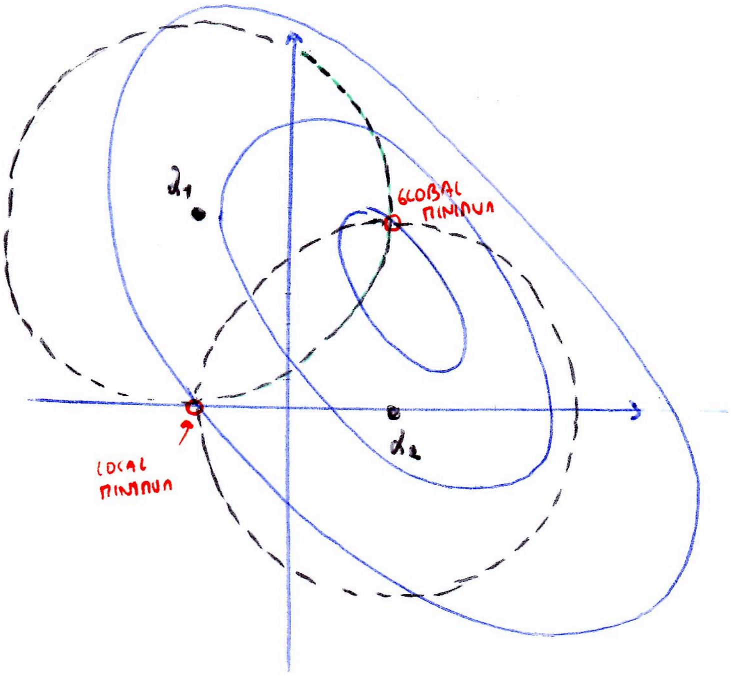
→ [Still a point in the Pareto frontiere]

[F(x) here are circles! not lines]

Back to Example 4: Antennas

if we use $F(\bar{X}) = \sum_{i=1}^{4} c_i \cdot d_i$ with $d_i = \sqrt{(x_1 - x_i)^2 + (x_2 - y_i)^2}$

we can find: (simulation with computer)



Example 5: Resources Allocation

The manager of a car race arranges 5 pilots (I, ..., V) and 5 cars (A, ..., E). The winning group must have its team must participate to a race where the winning group must have minimum global time or global performance of pilots according to the car/pilot table

$i \rightarrow$ / $j \downarrow$	A	B	C	D	E
I	3	5	6	9	10
II	4	8	8	11	13
III	6	9	10	12	14
IV	8	10	10	15	16
V	13	13	17	18	20

<u>Variables</u>: defined as $x_{ij} = \begin{cases} \emptyset & \text{otherwise} \\ 1 & \text{car } (i) \text{ driven by pilot } (j) \end{cases}$

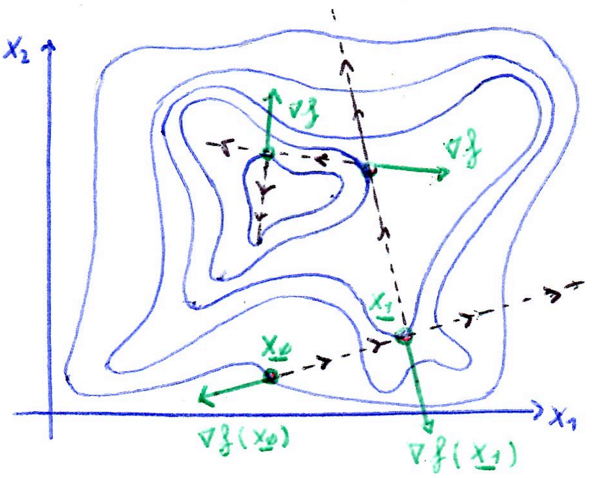
Example: $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{pmatrix} \text{I on A} \\ \text{II on B} \\ \text{etc...} \end{pmatrix}$

<u>Constraints</u>: $\sum_{i=1}^{J} x_{ij} = 1$; $\sum_{j=1}^{I} x_{ij} = 1$ ← (1 pilot ⟺ 1 car)

<u>Objective function</u>: $F(\bar{X}) = \sum_{j} \sum_{i} x_{ij} c_{ij}$ ← linear scalarization

E qui come finisco?

Steepest Descent Method (anticipation)



We start from $X_\emptyset = (X_\emptyset^{(1)}, X_\emptyset^{(2)})$

↓

∇f is in a direction ($\overrightarrow{\nabla f(X_\emptyset)}$ is a vector)

We move in the same direction with opposite sign ($\leftarrow X_\emptyset \dashrightarrow \dashrightarrow \dashrightarrow$)

until "it does not come up again" ($\rightarrow$ local minimum on the line)

Then we stop (X_1) and we do the same for $\overrightarrow{\nabla f(X_1)}$ (etc...)

↓

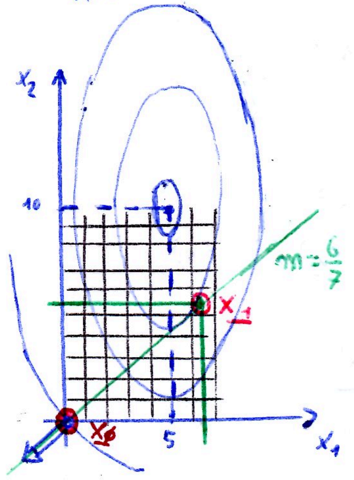
We changed a multivariable minimiz. problem in a monovariable minimization problem along the edges!

Example 6: Steepest Descent

We have $f(\bar{X}) = 200 + 7(x_1 - 5)^2 + 3(x_2 - 10)^2$

$$\nabla f = \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} = \begin{pmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \end{pmatrix} = \begin{pmatrix} 14(x_1 - 5) \\ 6(x_2 - 10) \end{pmatrix} \Rightarrow \bar{X}^* = \begin{pmatrix} 5 \\ 10 \end{pmatrix}$$ ← easy example

Let's prove this with gradient descent method ↓



Starting point: $\bar{X}_\emptyset = (\emptyset, \emptyset)^T$ (initial condition chosen like this)

↓

$$\nabla f(\bar{X}_\emptyset) = \begin{pmatrix} -70 \\ -60 \end{pmatrix}$$

We have now to find the equation of the line $D_\emptyset$ defined by $X_\emptyset$ and by the direction $\underline{d}_\emptyset = (7, 6)^T$

So, how we change from 2D to 1D?

~~1) $x_1 = x_2 = d \Rightarrow f(\bar{X}) = F(d) = 200 + 7(d-5)^2 + 3(x_2 - 10)^2$~~ no!

2) $\bar{X} = m x_1 + b$ and in this case $m = \frac{6}{7}$ and $b = \emptyset$

But is better in a parametric way:

3) $\boxed{\bar{X} = \bar{X}_\emptyset + d \cdot \vec{d}} \leftrightarrow (\overrightarrow{\bar{X}\bar{X}_\emptyset} = d\,\vec{d})$

descent method

---

o) $\nabla f(\bar{X}_\emptyset) = \begin{pmatrix} -70 \\ -60 \end{pmatrix} \Rightarrow \underline{d}_\emptyset = \begin{pmatrix} 7 \\ 6 \end{pmatrix}$ ... $f(\bar{X}_\emptyset + d\,\underline{d}_\emptyset) = F(d) = 200 + 7(7d - 5)^2 + 3(6d - 10)^2$

$F'(d) = 14(7d - 5)7 + 6(6d - 10)6 = 902d - 850 = \emptyset \Rightarrow d = \frac{850}{902} \simeq 0{,}94 \rightarrow \bar{X}_1 =$

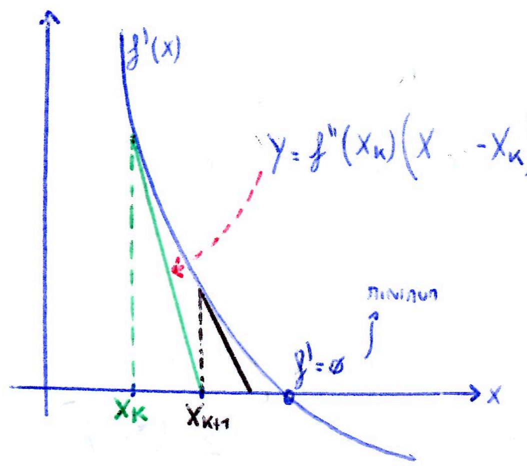
$$\hookrightarrow \bar{X}_1 = \begin{pmatrix} 6.58 \\ 5.64 \end{pmatrix}$$

so; $X_1 = \frac{850}{902}\begin{pmatrix}7\\6\end{pmatrix} = \begin{pmatrix}6.58\\5.69\end{pmatrix}$

∘) $\nabla f\begin{pmatrix}22.12\\-26.16\end{pmatrix} \cdot \begin{pmatrix}7\\6\end{pmatrix}$ CONTINUA A NON ESSERE ORTOGONALE

Single Variable Optimization (1)

1) **Newton Rapson Method**: THE OBJECTIVE FUNCTION HAS DERIVATIVES f' AND f''
↓
LET'S USE THEM



$$y = f''(x_k)(x - x_k) + f'(x_k)$$

[KNOWING THAT, TO HAVE THE TANGENT IN x_0 TO A FUNCTION $y(x)$ WE CAN COMPUTE:
$$y = y'(x_0)(x - x_0) + y(x_0)$$]

↳ TO HAVE x_{k+1}: $(y = \phi)$

$$y = \phi = f''(x_k)\cdot(x_{k+1} - x_k) + f'(x_k)$$

↓

$$\left[x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}\right]$$

Example 7: Newton Rapson

$f(x) = \frac{1}{3}x^3 - 2x^2 + 3x + 1$ $\quad x_0 = \phi \quad \Rightarrow \begin{cases} f'(x) = x^2 - 4x + 3 \\ f''(x) = 2x - 4 \end{cases}$

x_k	$f(x_k)$	$f'(x_k)$	$f''(x_k)$
START FROM ∅ $\boxed{\phi}$	1	3	-4
0.75	2.26	0.562	-2.5
0.975	2.33	0.05	-2.00
⋮	⋮	⋮	⋮
1	2.2	∅	$\boxed{< \phi}$

→ f'' GIVE US "MAXIMUM" $(< \phi)$ OR "MINIMUM" $(> \phi)$

↓

$f(1)$ IS MAXIMUM

IF WE START FROM $X_\emptyset = 4$,

X_K	$f(X_K)$	$f'(X_K)$	$f''(X_K)$
$\boxed{4}$	2.33	3	-4
3.25	1.067	0.562	-2.5
3.025	1	0.05	-2.00
⋮	⋮	⋮	⋮
3	1	$\emptyset$	$\boxed{<\emptyset}$

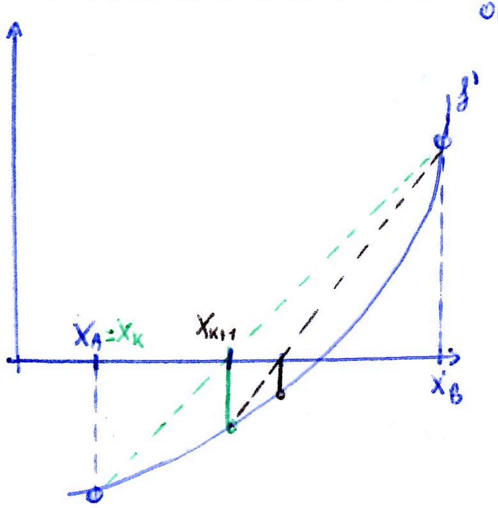
→ $f(3)$ IS A MINIMUM

Single Variable Optimization (2)

1) Newton-Raphson Method (cont'd): SO, AS WE SAW IN THE EXAMPLE, f'' GIVES US THE KIND OF POINT: $[f'' > \emptyset \Rightarrow$ MINIMUM$]$, $[f'' < \emptyset \Rightarrow$ MAXIMUM$]$!

TO STOP THE ITERATIONS WE CAN HAVE DIFFERENT STOPPING RULES, FOR EXAMPLE: $|f(X_{k+1}) - f(X_k)| < \varepsilon$; $f' < \varepsilon$; $|X_{k+1} - X_k| < \varepsilon$; etc...

2) Secant Iterative Method: THE APPROXIMATION OF THE MINIMUM CAN BE OBTAINED BY USING:

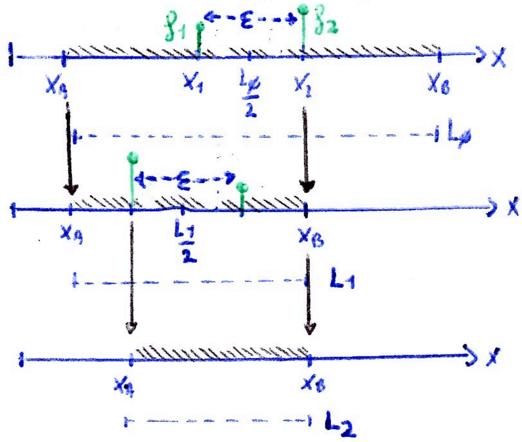


$$X_{K+1} = X_K - \frac{f'(X_K)}{\Delta}$$

WITH $$\Delta = \frac{f'(X_B) - f'(X_A)}{X_B - X_A}$$

[GOOD 'CAUSE YOU DON'T HAVE TO COMPUTE THE 2ND DERIVATIVE]

3) Dichotomous Search



IT WORKS WHEN OUR DESIRED MINIMUM IS IN $[X_A, X_B]$ AND NO OTHERS.

$$X_1 = \frac{L_\emptyset - \varepsilon}{2} \qquad X_2 = \frac{L_\emptyset + \varepsilon}{2}$$

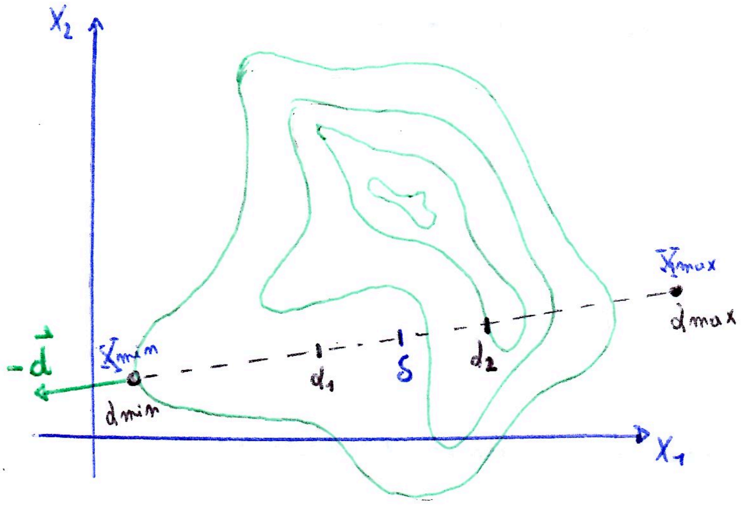
```
if (f(x1) ≜ f1) < (f(x2) ≜ f2)
    then XB ← X2
else     XA ← X1
end
```

Loop

$$L_1 = \frac{1}{2} L_\emptyset + \varepsilon$$

$$L_n = \frac{L_\emptyset}{2^{n/2}} + \varepsilon\left(1 - \frac{1}{2^{n/2}}\right)$$

Extension of Dichotomous Search to Multivariable f



$\left[\begin{array}{l} \bar{X}_{min}\ (\text{new } X_A) \\ \bar{X}_{max}\ (\text{new } X_B) \\ \delta \text{ indicates } \frac{L_n}{2} \text{ point} \end{array}\right]$ ↙ Better notation

After defining the 1D restriction (see gradient method) from X_{min}

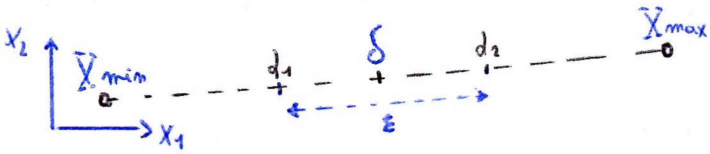
$$\bar{X}_{max} = \bar{X}_{min} + d \cdot \vec{d}$$

(indicates the direction) ⇒

$$\left[\overrightarrow{X_{min} X_{max}} = d\,\vec{d}\right]$$

To simplify, let's define the parameter d s.t.:

$$\bar{X} = \bar{X}_{min}\ (\Leftarrow d = \emptyset) \Rightarrow d_{min} = \emptyset \quad \Rightarrow L = d_{max} - \cancel{d_{min}} = d_{max}$$



$$\delta = \bar{X}_{min} + \frac{d_{max}}{2}\vec{d}$$

$$d_1 = \frac{d_{max}}{2} - \frac{\varepsilon}{2} \ ; \quad \bar{X}_1 = \bar{X}_{min} + d_1\vec{d}$$

$$d_2 = \frac{d_{max}}{2} + \frac{\varepsilon}{2} \ ; \quad \bar{X}_2 = \bar{X}_{min} + d_2\vec{d}$$

Unconstrained Multi-Variable Optimization (1)

Iterative descent method:

1.) From starting point (initial) $\bar{X}_\emptyset$

2.) Construct the improving design vector of points $(\bar{X}_1, \bar{X}_2, \dots, \bar{X}^*)$

For each point $\bar{X}_K$ find:

a.) A suitable direction $\vec{d}_K$

b.) An appropriate step length d_K along direction $\vec{d}_K$ (d_K can be fixed or evaluated each iteration)

3.) Find the approximation of $\bar{X}_{K+1} = \bar{X}_K + d_K\,\vec{d}_K$

4.) Test the descent condition $f(\bar{X}_{K+1}) < f(\bar{X}_K)$

5.) Test wether $\bar{X}_{K+1}$ is the optimum using stopping condition

Stopping Conditions:

a) Design variables:

- Absolute test : $\| \bar{X}_{k+1} - \bar{X}_k \| < \varepsilon_{x_a}$

- Relative test : $\dfrac{\| \bar{X}_{k+1} - \bar{X}_k \|}{\| \bar{X}_k \|} < \varepsilon_{x_r}$

example:

$X_{k+1} = 0.00023 = 2,3 \cdot 10^{-5}$

$X_k = 0.00022 = 2,2 \cdot 10^{-5}$

$\downarrow$

Absolute => 10^{-6}

Relative => 0.045

b) Objective Function:

- Absolute test: $| f(\bar{X}_{k+1}) - f(\bar{X}_k) | < \varepsilon_{f_a}$

- Relative test : $\dfrac{| f(\bar{X}_{k+1}) - f(\bar{X}_k) |}{| f(\bar{X}_k) |} < \varepsilon_{f_r}$

c) Derivate Function :

for example $\max\limits_{i=1,\dots,n} \left| \partial f / \partial x_i \right| < \varepsilon_d$

Descent Methods

As we saw, descent methods are iterative:

1) Find a direction $\underline{d}_k$ s.t. $\underline{d}_k^T \cdot \nabla f(\bar{X}_k) < \phi$

2) Find an optimal step α_k s.t. $f(\bar{X}_k + \alpha_k \underline{d}_k) < f(\bar{X}_k)$

3) Calculate $\bar{X}_{k+1} = \bar{X}_k + \alpha_k \underline{d}_k$

intuitive examples for the choice of α_k and $\underline{d}_k$:

- $\underline{d}_k = -\underline{\underline{D}}_k \cdot \nabla f(\bar{X}_k)$ with $\underline{\underline{D}}_k$ positive definite $\left(\Rightarrow -\nabla f(\bar{X}_k)^T \cdot D_k^T \nabla f(\bar{X}_k) < \phi \right)$

- $\hookrightarrow \underline{\underline{D}}_k = \underline{\underline{I}}$ -> Steepest Descent Method (seen before)

- α_k s.t. $\alpha_k = \min F(\alpha_k)$ with $F(\alpha_k) = f(\bar{X}_k + \alpha_k \underline{d}_k)$

Steepest Descent Method (Reprise)

We anticipated the method in 5. This approach is based on the fact that $\{-\nabla f(\bar{X}_k)\}$ represent the steepest direction at $\bar{X}_k$

→ example ([illegible]):

$$f(\bar{X}) = X_1 - X_2 + 2X_1^2 + 2X_1X_2 + X_2^2$$

starting point: $\bar{X}_\emptyset = (\emptyset, \emptyset)^T$

$$\nabla f(\bar{X}) = (1 + 4X_1 + 2X_2 ; -1 + 2X_1 + 2X_2)^T$$

(we could use dicotomy above here for example)

rewrite f in monodimensional version!

•) $\nabla f(\bar{X}_\emptyset) = (1, -1)^T \Rightarrow \underline{d} = (-1, +1)^T \qquad (\bar{X}_\emptyset + \lambda \underline{d}) = F\begin{pmatrix} -\lambda \\ \lambda \end{pmatrix} = -\lambda - \lambda + \cancel{2\lambda^2} - \cancel{2\lambda^2} + \lambda^2 = \lambda^2 - 2\lambda$

$\hookrightarrow dF/d\lambda = \underbrace{2\lambda - 2 \overset{!}{=} \emptyset}_{\text{monodimensional minimization problem}} \Rightarrow \lambda = 1 \quad \leadsto \bar{X}_1 = \bar{X}_\emptyset + \begin{pmatrix} -1 \\ 1 \end{pmatrix} = (-1, 1)^T$

•) $\nabla f(\bar{X}_1) = (1 - 4 + 2 ; -1 - 2 + 2) = (-1, -1) \qquad \underline{d}_1 = (1, 1)^T$

$$f\begin{pmatrix} -1+\lambda \\ 1+\lambda \end{pmatrix} = F(\lambda) = -1 + \lambda \cancel{-1} - \lambda + 2 + 2\lambda^2 - 4\lambda + 2\lambda^2 \cancel{-2} + \cancel{1} + \lambda^2 + 2\lambda = 5\lambda^2 - 2\lambda - 1$$

$$\frac{dF}{d\lambda} = 5\lambda - 1 \Rightarrow \lambda = \frac{1}{5} \quad \leadsto \bar{X}_2 = \begin{pmatrix} -1 + 1/5 \\ 1 + 1/5 \end{pmatrix} = \begin{pmatrix} -0.8 \\ 1.2 \end{pmatrix}$$

$$\bar{X}_3 = \begin{pmatrix} -1 \\ 1,4 \end{pmatrix} \to (\dots) \to \bar{X}^* = \begin{pmatrix} -1 \\ 1,5 \end{pmatrix} \xleftarrow{\text{for this}} \nabla f(\bar{X}^*) = (\emptyset, \emptyset)^T \to \text{minimum found!}$$

⚠ $\underline{d}_k^T \underline{d}_{k+1}$ ↳ in fact, since λ_k is the minimum of $F(\lambda) = f(\bar{X}_k + \lambda \underline{d}_k)$

one has:

$$\frac{dF}{d\lambda} = \underline{d}_k^T \cdot \nabla f(\bar{X}_k + \lambda_k \underline{d}_k) = \underline{d}_k^T \underbrace{\nabla f(\bar{X}_{k+1})}_{-\underline{d}_{k+1}} = \emptyset$$

Gradient Conjugate Method (Powell Method)

The optimum of a quadratic function $f(\bar{X}) = \frac{1}{2} \bar{X}^T \underline{A} \bar{X} + \underline{b} \bar{X} + c$ requires only n iterations! of Powell method.

$$\begin{bmatrix} n = \dim(\bar{X}) \\ n \times n = \dim(\underline{A}) \\ n = \dim(\underline{b}) \\ c = \text{cost.} \end{bmatrix}$$

directions $\underline{d}_i$ are $\underline{A}$-conjugate $\Leftrightarrow \left[\underline{d}_i^T \underline{A} \underline{d}_j = \emptyset \quad \forall i, j, i \neq j \right]$

(if A = Identity → conjugate = orthogonal)

THE ITERATIVE PROCESS:

$g_k = \nabla f(\bar{X}_k) = A\bar{X}_k + b^T$

FROM STARTING POINT $\bar{X}_\emptyset$ AND $\underline{d}_\emptyset = -g_\emptyset$:

a) $\bar{X}_{k+1} = \bar{X}_k + \alpha_k \underline{d}_k$, WITH $\left[\alpha_k = -\dfrac{g_k^T \underline{d}_k}{\underline{d}_k^T \underline{A} \underline{d}_k} \right]$

b) Then restart WITH $\left[\underline{d}_{k+1} = -g_{k+1} + \dfrac{g_{k+1}^T \underline{A} \underline{d}_k}{\underline{d}_k^T \underline{A} \underline{d}_k} \underline{d}_k \right]$

- example:

$f(\bar{X}) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \Rightarrow [A, b, c?] \Rightarrow f(\bar{X}) = \frac{1}{2}(x_1\ x_2)\left(? \right)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (1, -1)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$(c = \emptyset)$

STARTING POINT: $\bar{X}_\emptyset = (\emptyset, \emptyset)$

WE TAKE A FROM ∇f

$\nabla f = (1 + 4x_1 + 2x_2 , -1 + 2x_1 + 2x_2)^T$

$= \underbrace{\begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}}_{A}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \underbrace{\begin{pmatrix} 1 \\ -1 \end{pmatrix}}_{b^T} = g_k$

NOW WE CAN START

•) $g_\emptyset = (1, -1)^T$ $\quad \underline{d}_\emptyset = -g_\emptyset = (-1, 1)$ $\quad \alpha_\emptyset = \dfrac{-g_\emptyset^T \underline{d}_\emptyset}{\underline{d}_\emptyset^T \underline{A} \underline{d}_\emptyset} = 1$ $\quad f(\bar{X}_\emptyset) = \emptyset$

$\bar{X}_1 = \bar{X}_\emptyset + \alpha_\emptyset \underline{d}_\emptyset = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

•) $g_1 = \nabla f(\bar{X}_1) = \underline{A}\bar{X}_1 + \underline{b}^T = (-1, -1)^T$ $\quad \underline{d}_1 = -g_1 + \dfrac{g_1^T A \underline{d}_\emptyset}{\underline{d}_\emptyset^T A \underline{d}_\emptyset} \underline{d}_\emptyset = (\emptyset, 2)^T$

$\alpha_1 = \dfrac{-g_1^T \underline{d}_1}{\underline{d}_1^T A \underline{d}_1} = 0.25$ $\quad f(\bar{X}_1) = -1$ $\quad \bar{X}_2 = \bar{X}_1 + \alpha_1 \underline{d}_1 = (-1, 1.5)$

•) $g_2 = (\emptyset, \emptyset)^T \rightarrow$ OK $\quad X^* = (-1, 1.5)^T$ MINIMUM!

UNCONSTRAINED MULTI-VARIABLE OPTIMIZATION (2)

Heuristic Methods: THESE METHODS AVOID THE ∇f COMPUTATION

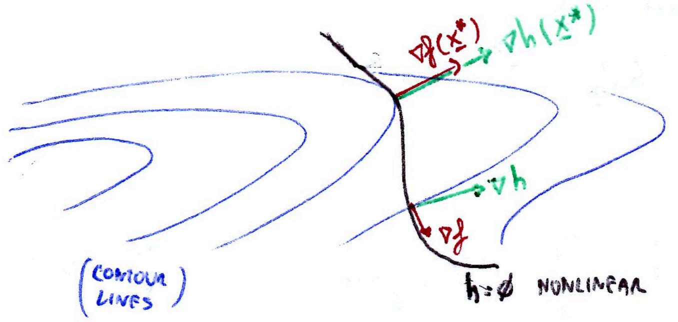
example: Hook and Jeeves → [PAR. 6. chapter 4 ON SLIDES]

Geometric Interpretation of Optimality Conditions (Syntesis of Chap. 5 slides)

Constrained optimization problem: $\left[\text{Find } \underline{x}^* = \min f(\underline{x}) \quad \text{where } \underline{x}^* \in \Omega \right]$

Ω : domain of admissible points: $\left[\underline{x} \text{ s.t. } \begin{cases} h_i(\underline{x}) = \emptyset & \forall i = 1, \dots, r \\ g_i(x) \leq \emptyset & \forall i = r+1, \dots \end{cases} \right]$

→ $h_i(\underline{x}) = \emptyset$ analysis:



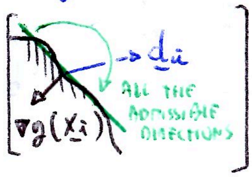
Sliding along the equality constraint, when we have $\nabla f \parallel \nabla h$ we cannot improve anymore our optimization:

$\left[x^* \text{ local minimum} \Leftrightarrow \nabla f(\underline{x}^*) \parallel \nabla h(\underline{x}^*) \right]$

→ $g_i(\underline{x}) \leq \emptyset$ analysis:

$\underline{d}_i$ admissible if $\underline{x}_{i+1} = \underline{x}_i + \lambda_i \underline{d}_i$ "for a certain λ" → $\left[\text{if } \exists \eta \text{ s.t. } \lambda \in [0, \eta] \Rightarrow \underline{x}_{i+1} \in \Omega \right]$

if $\left[\nabla g(\underline{x}_i)^T \cdot \underline{d}_i \leq \emptyset \right]$ ("if $\underline{x}_{i+1}$ is interior"

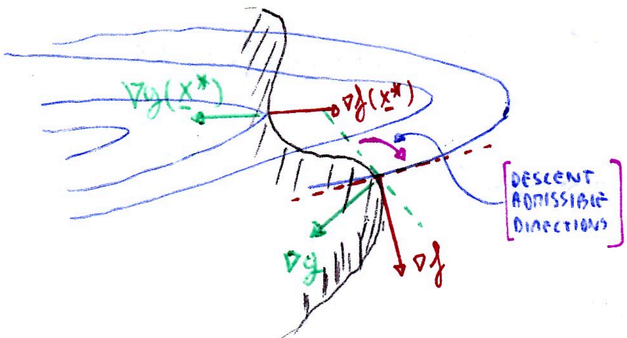


)

If f has a local minimum $\underline{x}^*$ then: $\left[\underline{d}^T \cdot \nabla f(\underline{x}^*) \geq \emptyset \right]$ $\forall$ admissible $\underline{d}$

g is an active constraint if $\left[g(x^*) = \emptyset \right]$

↳ so now we can say:

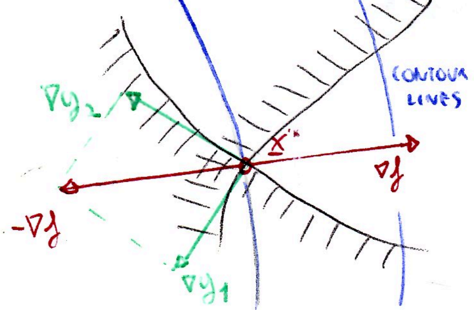


- Admissible: $\nabla g(\underline{x})^T \underline{d} < \emptyset$
- Descent: $\nabla f(\underline{x})^T \underline{d} < \emptyset$

$$\left[x^* \text{ local minimum} \Leftrightarrow \nabla f(\underline{x}^*) = -\underbrace{\lambda}_{\text{(real number)}} \nabla g(x^*) \right]$$

"they are $\parallel$ and opposite)

↳ If the current point is on two inequality constraints:



$$\exists \begin{matrix} \lambda_1 \geq \emptyset \\ \lambda_2 \geq \emptyset \end{matrix} \text{ s.t. } \lambda_1 \cdot \nabla g_1(\underline{x}) + \lambda_2 \cdot \nabla g_2(\underline{x}) = -\nabla f(\underline{x})$$

LAGRANGE FORMULATION (SYNTHESIS Chap.5 SLIDES)

LAGRANGE ALTERNATIVE FORMULATION TO OPTIMIZATION PROBLEM:

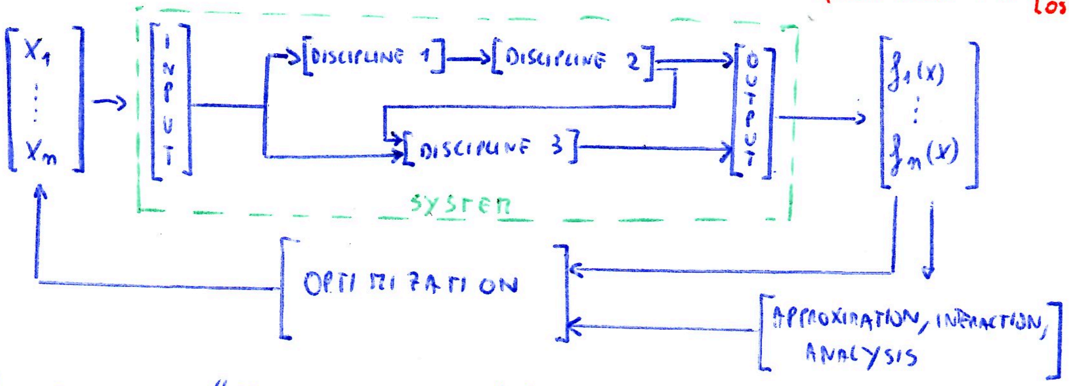
$$L(\bar{X}, \underline{\lambda}, \underline{\mu}) = f(\bar{X}) + \sum_{i=1}^{p} \mu_i \cdot h_i(\bar{X}) + \sum_{i=p+1}^{m} \lambda_i \cdot g_i(\bar{X})$$

n = # VARIABLES $\qquad$ m = # TOTAL CONSTRAINTS

THE NECESSARY CONDITION TO HAVE AN OPTIMUM IS $\nabla L(\bar{X}^*, \underline{\lambda}^*, \underline{\mu}^*) = \emptyset$

(SEE KUHN-TUCKER NECESSARY & SUFFICIENT CONDITION IN PAR. 6 OF CHAP. 5) → [AND Pg. 12 & 13]!

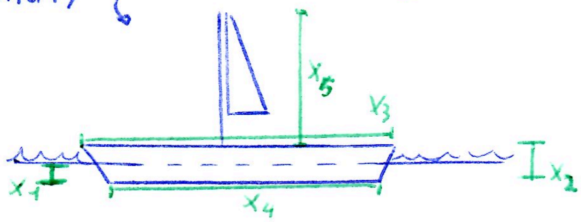
MULTI OBJECTIVE & MULTI DISCIPLINE OPTIMIZATION (FROM SLIDES THAT ARE LOST)



INTRODUCED THE "APPROX., INTERACT., AN." BLOCK BECAUSE SOMETIMES THE RAW OPTIMIZATION NEEDS TOO MUCH TIME

- example: COST-BASED OPTIMIZATION OF THE SHAPE OF A SHIP

[FOR A SHIP WE CAN HAVE DIFFERENT DESIGN PARAMETERS. TO SIMPLIFY]



[AND WE CAN HAVE DIFFERENT FUNCTION TO OPTIMIZE:]

$f_1(\underline{x})$: energy wave

$f_2(\underline{x})$: hydrodinamic

$f_3(\underline{x})$: Structural

$f_4(\underline{x})$: Cost

(SLIDES LOST...)

Simulated Annealing Algorithm

→ INSPIRED BY EXPERIMENTAL OBSERVATION ON CRYSTALIZATION FROM MELT.
→ AT THE HIGH TEMPERATURE THE ATOMS IN THE MELT ARE FREE TO MOVE AROUND THE SAMPLE.
→ AT REDUCED TEMPERATURE THE ATOMS TENDS TO CRYSTALIZE INTO A SOLID.
→ IF THE SAMPLE IS COOLED RAPIDLY THEN THE SOLID IS USUALLY POLYCRYSTALLINE.
→ IF THE SAMPLE IS COOLED SLOWLY (ANNEALED) THEN THE SOLID STANDS A BETTER CHANCE OF FORMING A PERFECT CRYSTAL
↳ GLOBAL OPTIMAL !

- General Structure of Simulated Annealing:

(Start With a System at a Known Configuration (Known energy E))
T = "hot" (Temperature set with a big value)
Frozen = "false" (Boolean value for Frozen state)
while (not Frozen)
  repeat
    PERTURB THE SYSTEM SLIGHTLY
    COMPUTE ΔE (CHANGE IN THE ENERGY DUE TO THE PERTURBATION)
    if ($\Delta E < \emptyset$)
      WE ACCEPT THE PERTURBATION
      THIS IS THE NEW CONFIGURATION
    else
      WE "MAYBE" ACCEPT IT WITH A PROBABILITY OF $\exp(-\Delta E / KT)$
  until (the system is in thermal equilibrium at this T)
  if (ΔE still decreasing over the last few temperatures)
    $T := a \cdot T$ $(a < 1)$ (COOL THE TEMPERATURE, DO PERTURBATION)
  else
    Frozen = "true"
RETURN THE FINAL CONFIGURATION AS Low Energy Solution

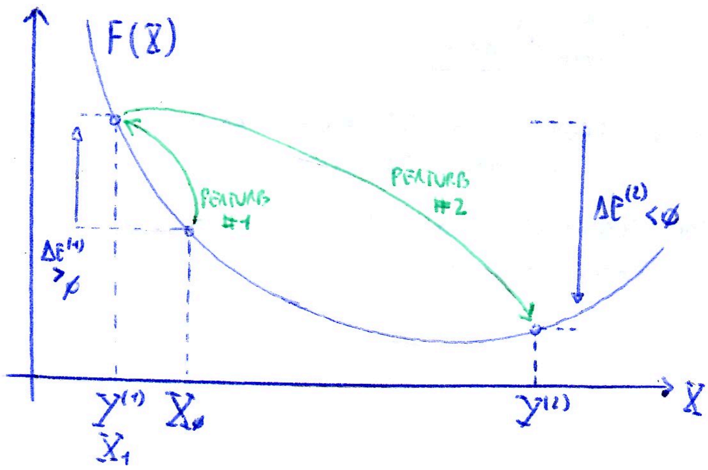
- Example of a S.A. Algorithm:

START WITH: [T = (INITIAL TEMPERATURE); T_f = (FINAL TEMP.); MaxIter = (Max number of iterations);
$X = X_\emptyset$ (INITIAL VALUE OF DESIGN VARIABLES); iter = 1 (ITERATION COUNTER)
$X_{opt} = X$ (STARTING "OPTIMAL" VALUE); $f_{opt} = f(X_{opt})$]

while ($T > T_f$)
  while (iter < MaxIter)
    iter ++
    CHOOSE Y CLOSE TO X AND COMPUTE $\Delta E = E(Y) - E(X)$
    if ($\Delta E < \emptyset$) (IF Y IS BETTER THAN X)
      $X = Y$
      if ($f(X) < f_{opt}$)
        $X_{opt} = X$ AND $f_{opt} = f(X)$
    else ($\Delta E \geq \emptyset$)

```
        else (ΔE ≥ 0)
            select "p" randomly in [0, 1]
            if (p < exp(-ΔE/T))      ← why no k?
                X = Y
    (end while)
    T: g(T)   (for example g(T) = aT with a < 1)
(end while)
return X_opt, f_opt
```

Let's apply the algorithm to this example:



- Starting with $X = X_0 = X_{opt}$

→ Perturb #1

$\Delta E^{(1)} > 0$ → pick a random p:

$p = 0.5$

T is "high", let's say $e^{-\Delta E/T} = 0.9$

So $0.5 = p < 0.9$ ↴

$Y^{(1)}$ is the new X (X_1)

→ Perturb #2

$\Delta E^{(2)} < 0$ so $Y^{(2)}$ is the new X

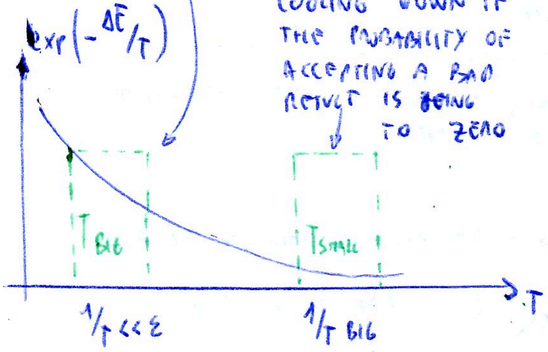
Check for minimum:

$F(X_2) < F_{opt} (= F(X_0))$

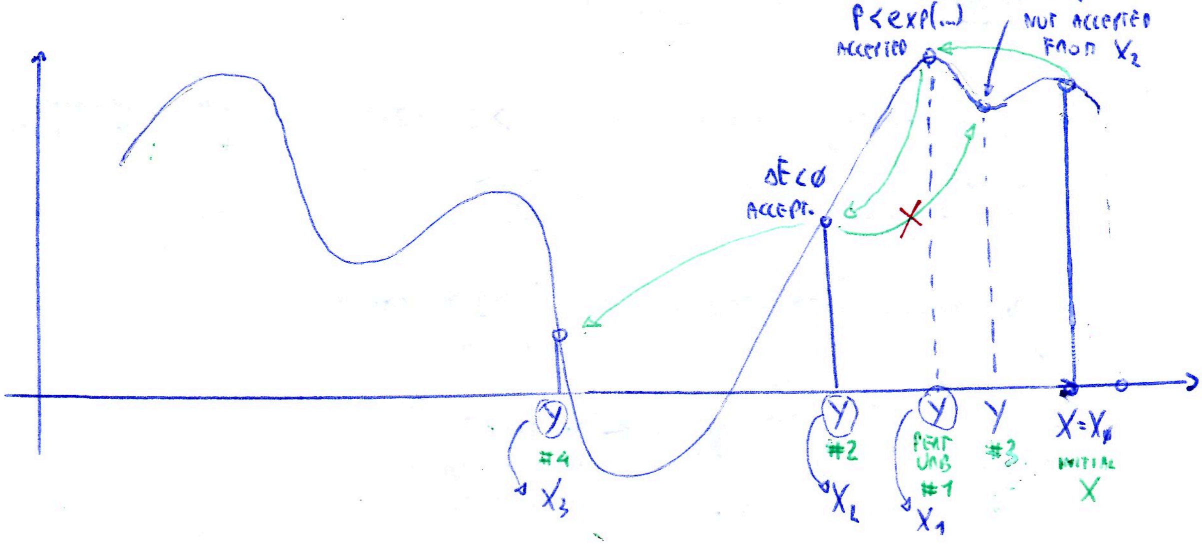
so $F_{opt} = F(X_2)$ ↴ start again...

The chance of accepting a "bad" result is high with high temperatures

Cooling down it the probability of accepting a bad result is going to zero



In a bigger example ↴



GENETIC ALGORITHMS

→ G.A. ARE DIRECTLY DERIVED FROM NATURAL EVOLUTION
→ G.A. MANTAIN AND MANIPULATE A FAMILY OR POPULATION
  ↳ G.A ARE POPULATION - BASED METHODS
→ G.A. ARE AN HEURISTIC APPROACH

- **TERMINOLOGY**: A POPULATION OF INDIVIDUALS IS EVOLVED TOWARDS THE SOLUTION
  AN INDIVIDUAL (OR "CHROMOSOME") IS A REPRESENTATION OF A POSSIBLE SOLUTION
  GENES: EACH INDIVIDUAL / CHROMOSOME IS MADE OF GENES,
  FOR EXAMPLE $(\emptyset, 1)$; $(A, B, C, \dots)$; INTEGERS ...

- **OPERATORS**: CROSSOVER: SELECTING TWO INDIVIDUALS (OR MORE) FOR THE POPULATION AND CREATING A CHILD, BUT KEEP SOME PROPERTIES OF THE 2 PARENTS

  MUTATION: TAKES AN INDIVIDUAL FROM THE POPULATION AND CHANGES SOME OF ITS GENES (A LITTLE)

  SELECTION: WHICH PARTS OF THE POPULATION ARE "BETTER" IS JUDGED ON THE FITNESS OF EACH INDIVIDUAL

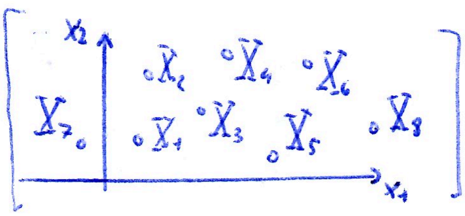
- **GENERAL ALGORITHM**:

```
initialize P(t=0) (POPULATION AT time 0)
t=1
while (not end)
    P(t) = Selection from [P(t-1)]   ← [SELECTING INDIVIDUALS FOR THE REPRODUCTION, DONE RANDOMLY WITH A PROBABILITY DEPENDING ON THE RELATIVE FITNESS OF THE INDIVIDUALS
                                         ↳ BEST ONES ARE MORE LIKELY CHOSEN]
    P(t) = Reproduction [P(t)]       ← [GENERATING NEW CHROMOSOMES WITH CROSSOVER AND MUTATION]
    P(t) = Evaluation [P(t)]         ← [EVALUATION OF FITNESS OF NEW CHROMOSOMES]
    P(t) = Replacement [P(t)]        ← [OLD INDIVIDUALS ARE KILLED AND SUBSTITUTED WITH THE NEW ONES]
    t = t+1
end
```

- **DESIGN VARIABLES**: $\underline{X} = [x_1, \dots, x_m]$ x_i are the GENES (m GENES FOR EACH INDIVIDUAL)
  INDIVIDUAL: $\underline{X}^{(k)} = [x_1^{(k)}, \dots, x_m^{(k)}]$ (N INDIVIDUALS)
  POPULATION AT TIME t: $P(t) = [\underline{X}^{(1)}(t), \dots, \underline{X}^{(N)}(t)] = [\underline{X}^{(1)}, \dots, \underline{X}^{(N)}]$

- **EXAMPLE WITH 2 VARIABLES**: $\underline{X} = [x_1, x_2]$ [N = SIZE OF POPULATION $> 2m = 2 \cdot$ NUMBER OF GENES FOR EACH INDIVIDUAL]
  THIS IS A GOOD CRITERIA TO FIND A SOUND SOLUTION
  $\underline{X}^{(k)} = [x_1^{(k)}, x_2^{(k)}]$ $P(t) = [\underline{X}^{(1)}, \dots,]$

  WE CHOOSE $N=8$ (RESPECTING THE $\underbrace{N}_{8} > \underbrace{2 \cdot m}_{4}$ CRITERIA)



← [AT TIME t THIS IS THE POPULATION]

- EXAMPLE OF SELECTION STRATEGY:

FOREACH INDIVIDUAL $\vec{X}^{(k)}(t-1)$ OF $P(t-1)$

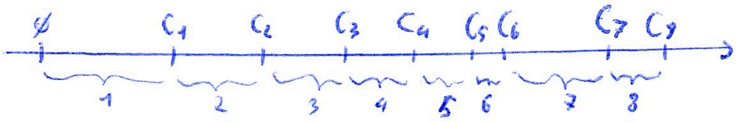
a) COMPUTE THE PROBABILITY OF SELECTION p_i (p_i REPRESENTS THE FITNESS OF THE INDIVIDUAL. IT IS EVALUATED W.R.T. THE OBJECTIVE FUNCTION IN ORDER TO CHOSE WITH HIGHER PROBABILITY THE BEST INDIVIDUALS)

FOR EXAMPLE: $p_i = F(\vec{X}^{(i)}) \Big/ \sum_{j=1}^{N} F(\vec{X}^{(j)})$

FITNESS

b) COMPUTE $C_i = \sum_{j=1}^{i} p_j$

example:



$$\begin{bmatrix} C_1 = p_1 \\ C_2 = p_1 + p_2 \\ C_3 = p_1 + p_2 + p_3 \\ (\dots) \end{bmatrix}$$

c) COMPUTE $r_i = U(\emptyset, 1)$ (IT'S A RANDOM UNIFORM VALUE IN $[0,1)$ INTERVAL)

IF $C_{j-1} < r_i \leq C_j$ THEN $\vec{X}^{(i)}(t) = \vec{X}^{(j)}(t-1)$

THE NEW i IS SELECTED AS THE OLD j

in the previous example: THE PROBABILITY OF PICKING $\vec{X}^{(1)}$ IS MORE THAN THE ONE OF PICKING $\vec{X}^{(6)}$

!!! EACH INDIVIDUALS CAN BE SELECTED MANY TIMES !!!

→ example $(N=8)$:

1) $r_1 = 0.6$ $\quad C_4 < r_1 < C_5$ $\quad \rightarrow \vec{X}^{(1)}(t) = \vec{X}^{(5)}(t-1)$
2) $r_2 = 0.2$ $\quad C_1 < r_2 < C_2$ $\quad \rightarrow \vec{X}^{(2)}(t) = \vec{X}^{(2)}(t-1)$
3) $r_3 = 0.9$ $\quad C_6 < r_3 < C_7$ $\quad \rightarrow \vec{X}^{(3)}(t) = \vec{X}^{(7)}(t-1)$
4) $r_4 = 0.5$ $\quad C_4 < r_4 < C_5$ $\quad \rightarrow \vec{X}^{(4)}(t) = \vec{X}^{(5)}(t-1)$

⋮

8) $r_8 = \dots$ $\quad \rightarrow P(t) = [\vec{X}^{(5)}, \vec{X}^{(2)}, \vec{X}^{(7)}, \vec{X}^{(5)}, \dots]$

- CROSSOVER:

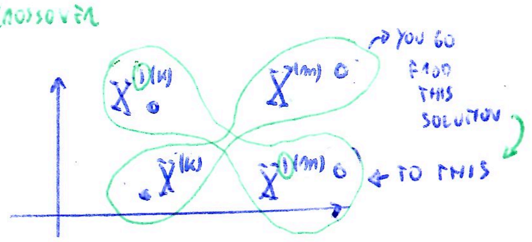
TAKE 2 INDIVIDUALS AND PRODUCES NEW INDIVIDUAL

$$\vec{X}^{(k)} = [x_1^{(k)}, \dots, x_{r-1}^{(k)}, x_r^{(k)}, \dots, x_m^{(k)}]$$
$$\vec{X}^{(m)} = [x_1^{(m)}, \dots, x_{r-1}^{(m)}, x_r^{(m)}, \dots, x_m^{(m)}]$$

→ COMPUTE $r \in [1, m]$ RANDOMLY

CROSSOVER

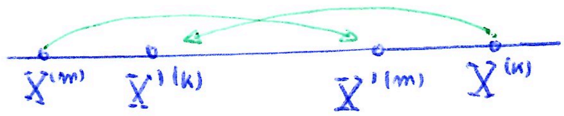
$$\vec{X}^{1(k)} = [x_1^{(k)}, \dots, x_{r-1}^{(k)}, x_r^{(m)}, \dots, x_m^{(m)}]$$
$$\vec{X}^{1(m)} = [x_1^{(m)}, \dots, x_{r-1}^{(m)}, x_r^{(k)}, \dots, x_m^{(k)}]$$



- **Arithmetic crossover:**

$$\vec{X}'^{(k)} = r\cdot\vec{X}^{(k)} + (1-r)\,\vec{X}^{(m)} = \vec{X}^{(m)} + r\cdot\left(\vec{X}^{(k)} - \vec{X}^{(m)}\right) \qquad \text{with } r \in U[0,1]$$

$$\vec{X}'^{(m)} = (1-r)\,\vec{X}^{(k)} + r\cdot\vec{X}^{(m)} = \vec{X}^{(k)} + r\cdot\left(\vec{X}^{(m)} - \vec{X}^{(k)}\right)$$



- **Mutation:**

For real-valued representations:

→ Uniform mutation: randomly selects a variable j and set it equal to a random $U[a_i, b_i]$

$$x_i = \begin{cases} U(a_i, b_i) & \text{if } i = j \\ x_i & \text{otherwise} \end{cases}$$

→ Bondary mutation:

$$x_i = \begin{cases} a_i & \text{if } i = j \wedge r < 0.5 \\ b_i & \text{if } i = j \wedge r \geq 0.5 \\ x_i & \text{otherwise} \end{cases}$$

There are other ways of mutation...

EXERCISES & LAST REVIEW (LOST LAST LECTURE)

- GRADIENT DESCENT (EXERCISE):

$f(x) = (x_1 - 4)^2 + (x_2 - 4)^2 + x_1 + x_2 + x_1 x_2 + 1$

$\vec{X}_\emptyset = \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \qquad \vec{X}_1 = ?$

$\nabla f = \left(\partial f/\partial x_1 , \partial f/\partial x_2 \right) = \left(2(x_1 - 4) + 1 + x_2 ; 2(x_2 - 4) + 1 + x_1 \right) = \left(2x_1 + x_2 - 7 ; x_1 + 2x_2 - 7 \right)$

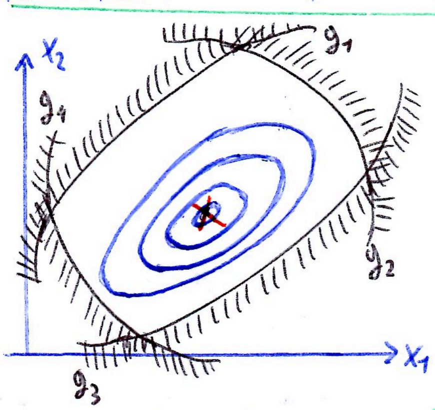
$d_\emptyset = -\nabla f(\vec{X}_\emptyset) = -(-7, -7)^T = (\cancel{7}, \cancel{7})^T \Rightarrow f(\vec{X}_\emptyset + \alpha \underline{d}_\emptyset) = f\begin{pmatrix} \alpha \\ \alpha \end{pmatrix} = F(\alpha)$

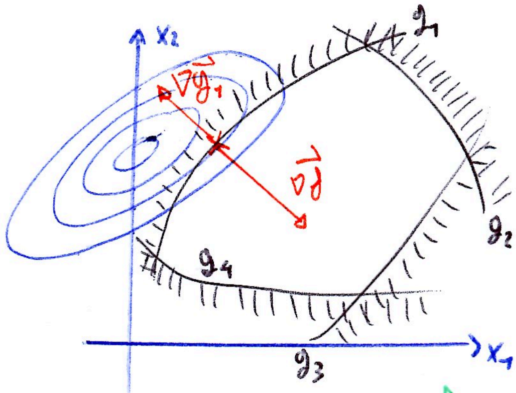
$F(\alpha) = (\alpha - 4)^2 + (\alpha - 4)^2 + \alpha + \alpha + \alpha^2 + 1 = 2(\alpha^2 + 16 - 8\alpha) + 2\alpha + \alpha^2 + 1 =$

$= 3\alpha^2 - 14\alpha + 33$

$F'(\alpha) = 6\alpha - 14 \overset{!}{=} \emptyset \rightarrow \alpha = \frac{14}{6} = \frac{7}{3} \rightarrow \vec{X}_1 = \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} + \begin{pmatrix} 7/3 \cdot 1 \\ 7/3 \cdot 1 \end{pmatrix} = \begin{pmatrix} 2.3 \\ 2.3 \end{pmatrix}$ ✓

- MULTIVARIABLE OPTIMIZATION WITH CONSTRAINTS (LAGRANGE FORMULATION)



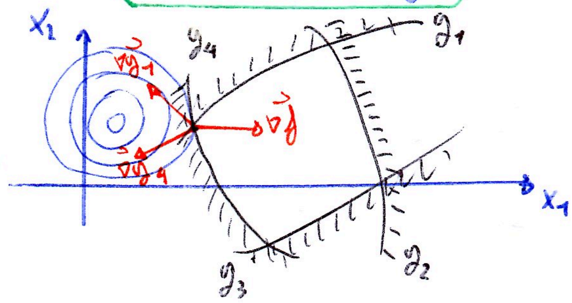


$\vec{\nabla} f = \vec{\emptyset} \Rightarrow x^*$ LOCAL MINIMUM REACHED

$\vec{\nabla} f = \lambda \, \nabla \vec{g}_1$

ACTIVE CONSTRAINT : g_1

(AS ALREADY SEEN IN 9)!



$\vec{\nabla} f = \lambda_1 \nabla \vec{g}_1 + \lambda_2 \nabla \vec{g}_4$

g_1 AND g_4 ACTIVE !

n = # VARIABLES $\qquad m$ = # CONSTRAINTS

[REWRITE h AND g AS ALL g "CONSTRAINTS" (EQUALITY & INEQUALITY) → EASY AND COMPACT] ↘

$L(\vec{X}, \lambda) = f(\vec{X}) + \sum_{i=1}^{m} \lambda_i \, g_i(\vec{X})$

NEC. COND. TO HAVE A MINIMUM IS $\nabla L(\vec{X}^*, \lambda^*) = \emptyset$

$$\left[\nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\vec{X}) = \emptyset \right]$$

AND THE SUFFICIENT? KUHN-TUCKER

⇓

(...)

- **KUHN-TUCKER CONDITIONS:**

[$\underline{X}^*$ is local minimum of $f(\underline{X})$

AND

REGULARITY CONDITIONS:

All active g_i are linear (LCQ)

OR

∇g_i of all active ∇g_i are lin. independ. (LICQ)

OR

(...) see wikipedia eng]

$\Longrightarrow$

[KKT ①]: STATIONARITY

$\nabla L(\underline{X}^*, \underline{\lambda}^*) = \emptyset \Leftrightarrow \nabla f(\underline{X}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\underline{X}^*) = \emptyset$

[KKT ②]: PRIMAL FEASIBILITY:

$g_i(\underline{X}) \leq \emptyset$ ($= \emptyset$ for equality constraints)

[KKT ③]: DUAL FEASIBILITY:

$\lambda_i^* \geq \emptyset$

② ③ "OBVIOUS"

[KKT ④]: COMPLEMENTARY SLACKNESS:

$\lambda_i^* g_i = \emptyset \begin{cases} \text{if } g_i \text{ inactive} \rightarrow \lambda_i^* = \emptyset \\ \text{if } g_i \text{ active} \rightarrow \lambda_i^* > \emptyset \end{cases}$

$\Longleftarrow$ [YOU NEED, IN GENERAL, KKT ④ TO HAVE SUFFICIENT CONDITION]

II ORDER KKT NEC. CONDITION

[KKT ⑤]:

$\underline{d}^T \underline{W}^* \underline{d} > \emptyset$, $\forall$ admissible $\underline{d}$

(where $\underline{W}^* = \nabla^2 L(\underline{X}^*, \underline{\lambda}^*)$)

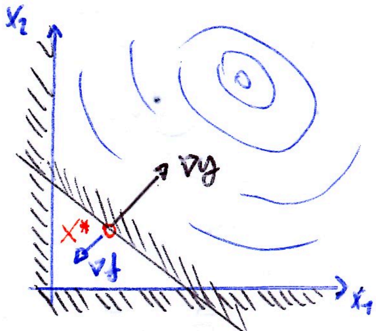
(① to ④ are FIRST-ORDER CONDITIONS)

- **KKT EXAMPLE:**

$f(\underline{X}) = (x_1 - 2)^2 + (x_2 - 2)^2$ $\qquad \nabla f(\underline{X}) = \begin{pmatrix} 2(x_1-2) \\ 2(x_2-2) \end{pmatrix}$

$\begin{cases} g_1(\underline{X}) = -x_1 \\ g_2(\underline{X}) = -x_2 \\ g_3(\underline{X}) = x_1 + x_2 - 1 \end{cases}$ $\qquad \nabla g_1 = (-1 \;\; \emptyset)^T \qquad \nabla g_2 = (\emptyset \;\; -1)^T \qquad \nabla g_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

to plot it? $x_2 = -x_1 + 1$



GRAPHICALLY WE CAN GUESS THAT g_3 IS ACTIVE ($g_3(\underline{X}^*) = \emptyset$) $\rightarrow$ see graph

CHECK KKT: $L = (x_1 - 2)^2 + (x_2 - 2)^2 - \lambda_1 x_1 - \lambda_2 x_2 + \lambda_3 (x_1 + x_2 - 1)$

①

$\nabla L = \emptyset \rightarrow \begin{cases} 2(x_1 - 2) - \lambda_1 + \lambda_3 = \emptyset \\ 2(x_2 - 2) - \lambda_2 + \lambda_3 = \emptyset \end{cases}$

②

$g_i \leq \emptyset \rightarrow \begin{cases} -x_1 \leq \emptyset \\ -x_2 \leq \emptyset \\ x_1 + x_2 - 1 \leq \emptyset \end{cases}$

③

$\lambda_i^* \geq \emptyset \rightarrow \begin{cases} \lambda_1^* \geq \emptyset \\ \lambda_2^* \geq \emptyset \\ \lambda_3^* \geq \emptyset \end{cases}$

② & ③ "OBVIOUS", USE TO CHECK RESULTS FROM ① AND ④

④

$\lambda_i^* g_i = \emptyset \rightarrow \begin{cases} \lambda_1^* g_1 = \emptyset \\ \lambda_2^* g_2 = \emptyset \\ \lambda_3^* g_3 = \emptyset \end{cases} \rightarrow \begin{cases} -\lambda_1 x_1 = \emptyset \; ; \text{ if } g_1 \text{ is active} \\ -\lambda_2 x_2 = \emptyset \; ; \text{ if } g_2 \text{ is active} \\ \lambda_3 x_1 + \lambda_3 x_2 - \lambda_3 = \emptyset \; ; \text{ if } g_3 \text{ is active} \end{cases}$

[if g_i inactive $\lambda_i^* = \emptyset$

if g_i active $\lambda_i^* > \emptyset$]

WE ITERATIVELY SOLVE EVALUATING ACTIVE CONSTRAINTS:

→ WE START WITH NO ACTIVE CONSTRAINTS

EQUATIONS FROM (1) $\begin{cases} 2x_1 - 4 - \lambda_1 + \lambda_3 = 0 \\ 2x_2 - 4 - \lambda_2 + \lambda_3 = 0 \end{cases}$

FROM (4) $g_i(X^*) \neq 0$ BECAUSE NOT ACTIVE SO, SINCE FOR KKT (3) $\lambda_i^* g_i(X^*) = 0 \Rightarrow \lambda_i^* = 0$

$\hookrightarrow \begin{cases} 2x_1 - 4 - 0 + 0 \\ 2x_2 - 4 - 0 + 0 \end{cases} \rightarrow \begin{cases} x_1 = 2 \\ x_2 = 2 \end{cases}$

CHECK w.r.t (2) AND (3):

(2) $\rightarrow \begin{cases} -2 \leq 0 & \text{OK} \\ -2 \leq 0 & \text{OK} \\ 2+2-1 \leq 0 & \text{NO}! \end{cases}$ → NOT GOOD

→ WITH g_3 ACTIVE:

(1) $\rightarrow \begin{cases} 2x_1 - 4 - \cancel{\lambda_1} + \lambda_3 = 0 \\ 2x_2 - 4 - \cancel{\lambda_2} + \lambda_3 = 0 \end{cases} \rightarrow \begin{cases} x_1 = 2 - \frac{1}{2}\lambda_3 \\ x_2 = 2 - \frac{1}{2}\lambda_3 \end{cases}$

(4) $\rightarrow \begin{cases} \lambda_1 = 0 \\ \lambda_2 = 0 \\ \lambda_3 x_1 + \lambda_3 x_2 - \lambda_3 = 0 \end{cases}$ $\rightarrow \left(2\lambda_3 - \frac{1}{2}\lambda_3^2\right)2 - \lambda_3 = -\lambda_3^2 + 3\lambda_3 = 0 \begin{cases} \lambda_3 = 0 \\ \lambda_3 = +3 \end{cases}$

$x_1 = 2 - \frac{1}{2}(+3) = \frac{1}{2}$ $\qquad x_2 = \frac{1}{2}$

CHECK w.r.t. (2) AND (3):

(2) $\rightarrow \begin{cases} -1/2 \leq 0 \\ -1/2 \leq 0 \\ \frac{1}{2} + \frac{1}{2} - 1 \leq 0 \end{cases}$ OK $\qquad$ (3) → OK $\qquad$ → OK → $X^* = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$ LOCAL MINIMUM ✓

- **KKT EXAMPLE:**

$f(X) = (x_1-2)^2 + (x_2-2)^2 \rightarrow \nabla f = \left(2(x_1-2);\ 2(x_2-2)\right)^T$

$\begin{cases} g_1(X) = -x_1 \\ g_2(X) = -x_2 \\ g_3(X) = x_1^2 + x_2^2 - 1 \end{cases} \rightarrow \begin{cases} \nabla g_1 = (-1; 0)^T \\ \nabla g_2 = (0; -1)^T \\ \nabla g_3 = (2x_1; 2x_2)^T \end{cases}$

DIFFERENCE W.R.T. BEFORE

$L(X, \lambda) = (x_1-2)^2 + (x_2-2)^2 - \lambda_1 x_1 - \lambda_2 x_2 + \lambda_3(x_1^2 + x_2^2 - 1)$

KKT (1) $\rightarrow \nabla L = 0 \begin{cases} 2(x_1-2) - \lambda_1 + \lambda_3(2x_1) \\ 2(x_2-2) - \lambda_2 + \lambda_3(2x_2) \end{cases}$ $\qquad$ KKT (4) $\rightarrow \lambda_i^* g_i = 0 \rightarrow \begin{cases} -\lambda_1^* x_1 = 0 & ;\ \text{IF } g_1 \text{ ACT} \\ -\lambda_2^* x_2 = 0 & ;\ \text{IF } g_2 \text{ ACT} \\ \lambda_3^*(x_1^2 + x_2^2 - 1) = 0 & ;\ \text{IF } g_3 \text{ ACT} \end{cases}$

KKT (2) $\rightarrow g_i \leq 0 \begin{cases} -x_1 \leq 0 \\ -x_2 \leq 0 \\ x_2^2 + x_2 - 1 \leq 0 \end{cases}$ $\qquad$ KKT (3) $\rightarrow \lambda_i^* \geq 0$

FINISH IT ooo