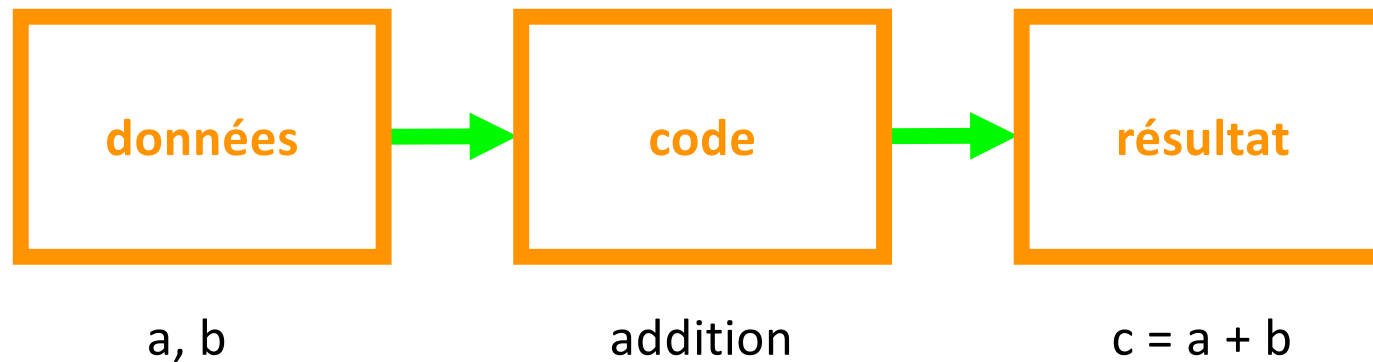


# Introduction aux algorithmes d'IA pour la recherche

# Codage traditionnel v.s. Intelligence artificielle

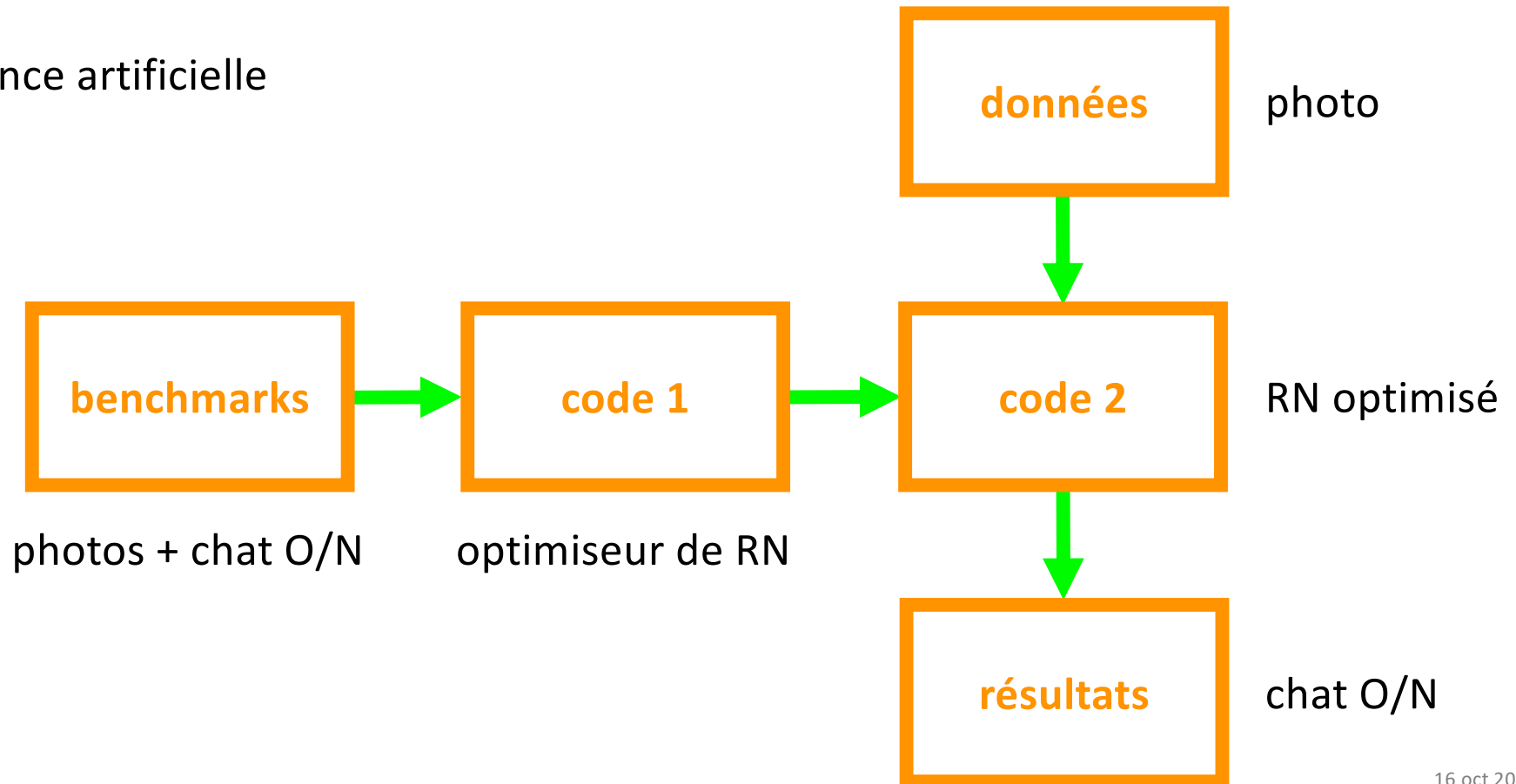
## Codage traditionnel



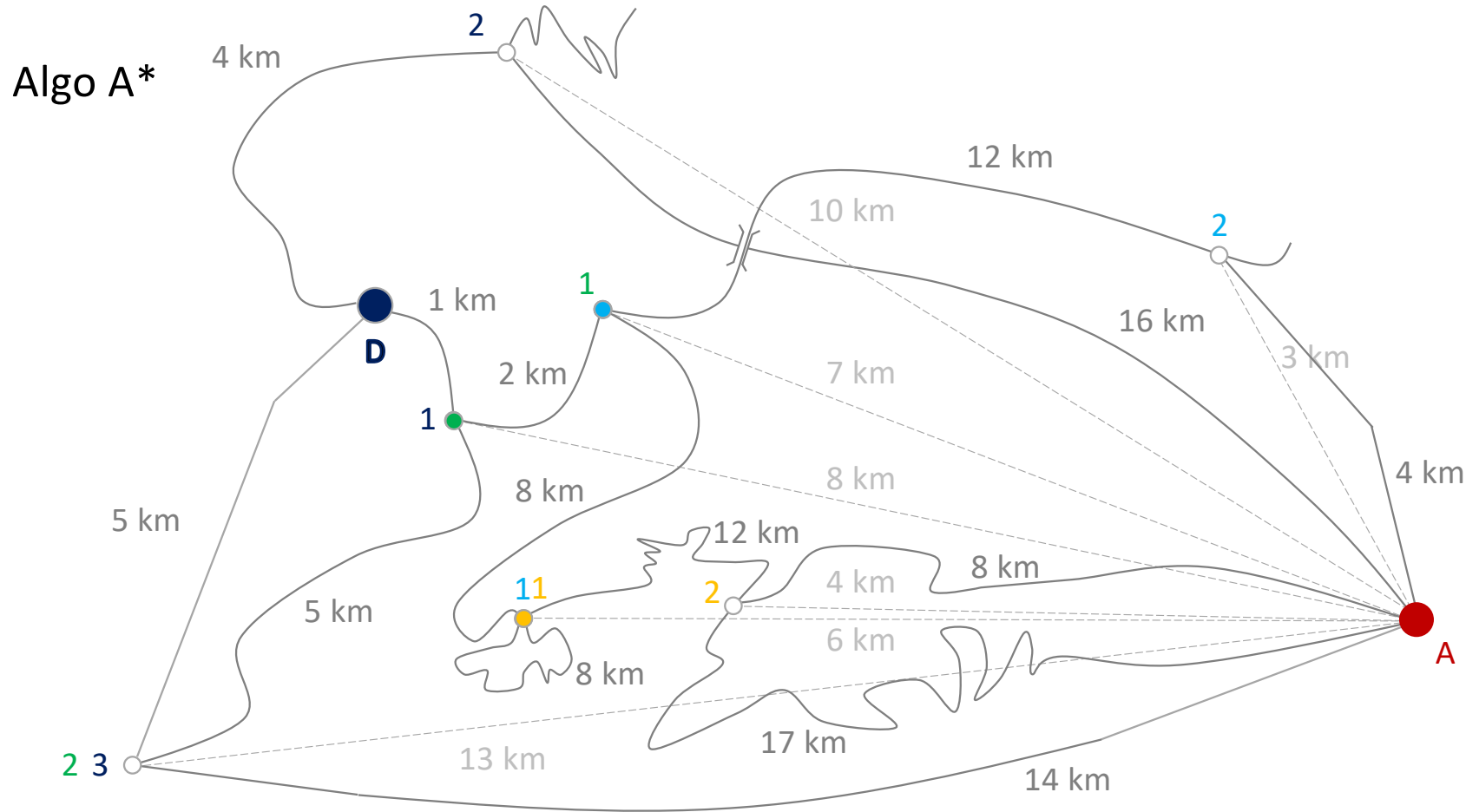


# Codage traditionnel v.s. Intelligence artificielle

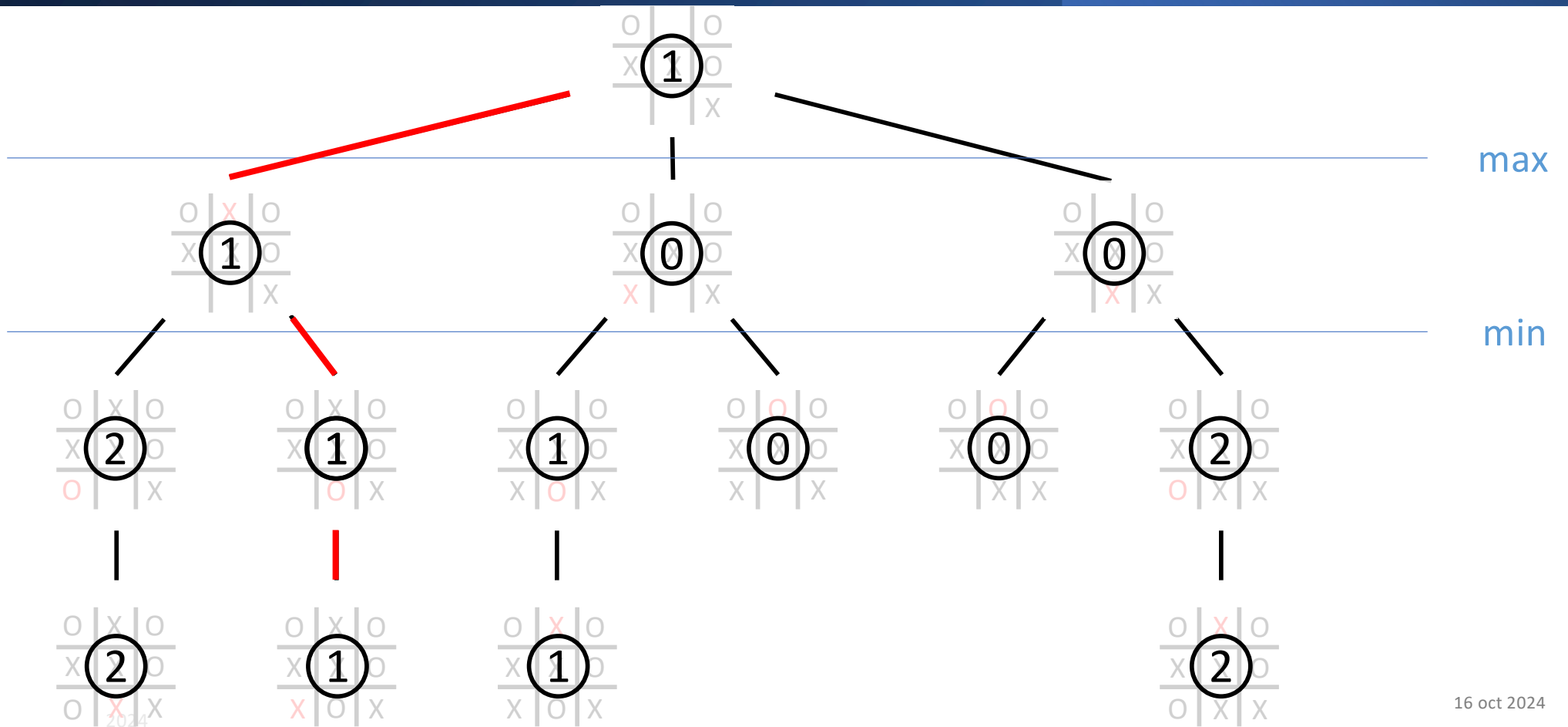
Intelligence artificielle



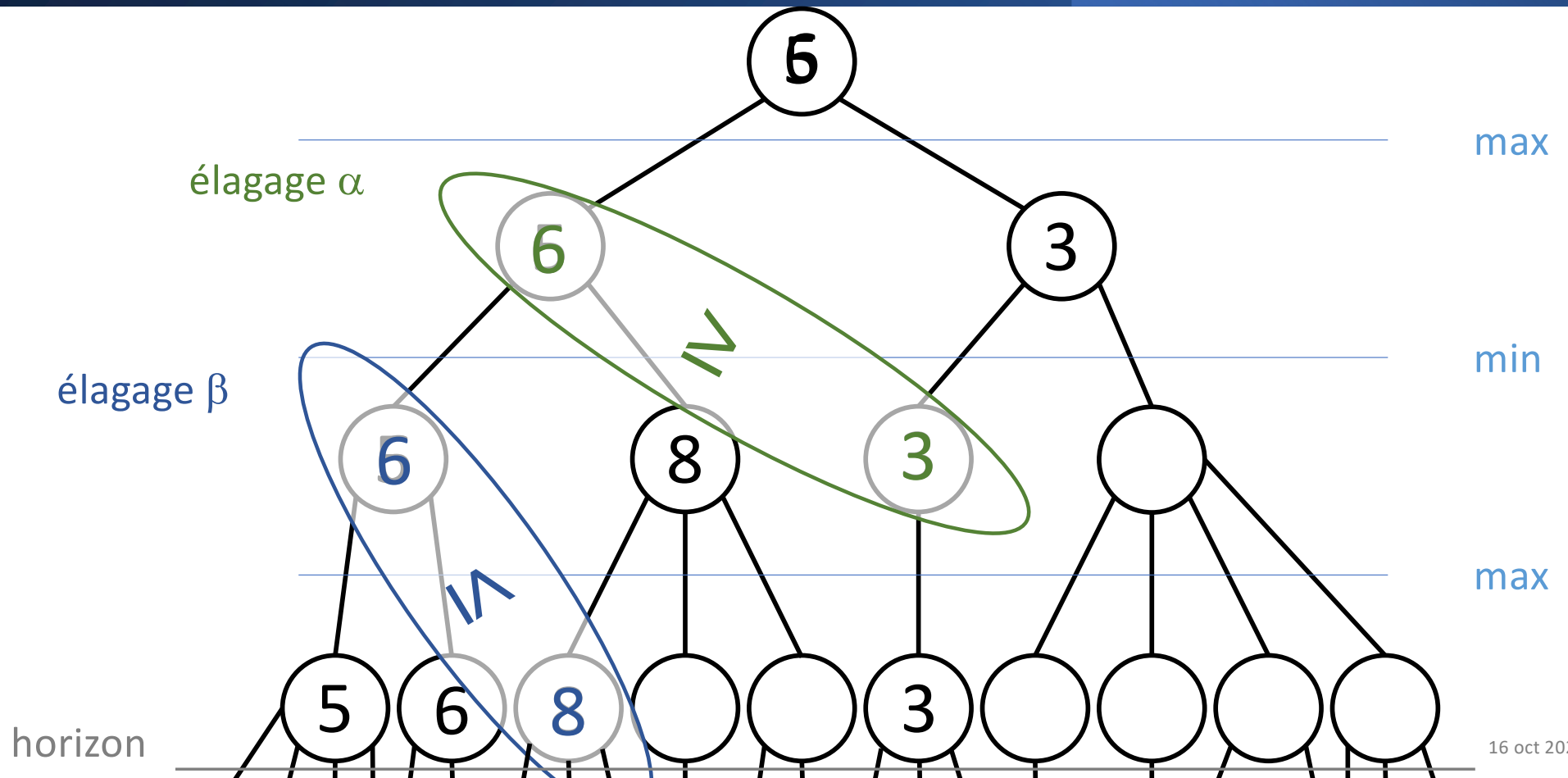
# heuristique



# Algorithme minmax



# Horizon, heuristique, élagage $\alpha$ $\beta$



# Algorithmes génétiques

## Le voyageur de commerce



Glouton

E
B
G
D
F
A
C

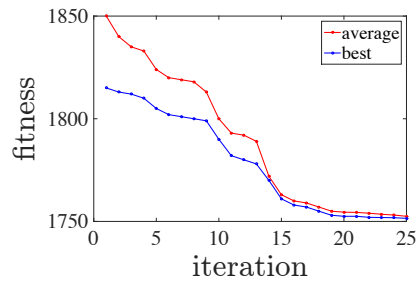
2280 km

Mon oeil

E
C
B
G
D
F
A

1800 km

# Algorithmes génétiques



stop

chromosomes

E	E	G	B	A	G
B	C	A	D	B	F
G	B	F	E	C	E
D	G	D	F	D	D
F	D	C	A	E	C
A	F	B	C	F	B
C	A	E	G	G	A

fonction d'évaluation

génération initiale  
(aléatoire)

E	G	G
C	A	F
B	F	E
G	D	D
D	C	C
F	B	B
A	E	A

sélection  
(n% meilleurs)

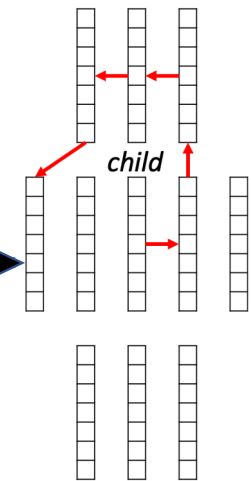
père mère enfant

G	G	G
F	A	F
E	F	E
D	D	A
C	C	D
B	B	C
A	E	B

croisement  
(enjambement)

G
C
E
A
D
F
B

mutation  
(très rares)



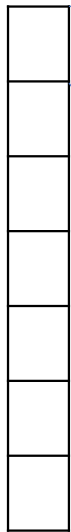
algorithme  
mémétique

itérations

# Algorithmes génétiques : le pilotage

chromosome

gène



⋮

distance  
et vitesse



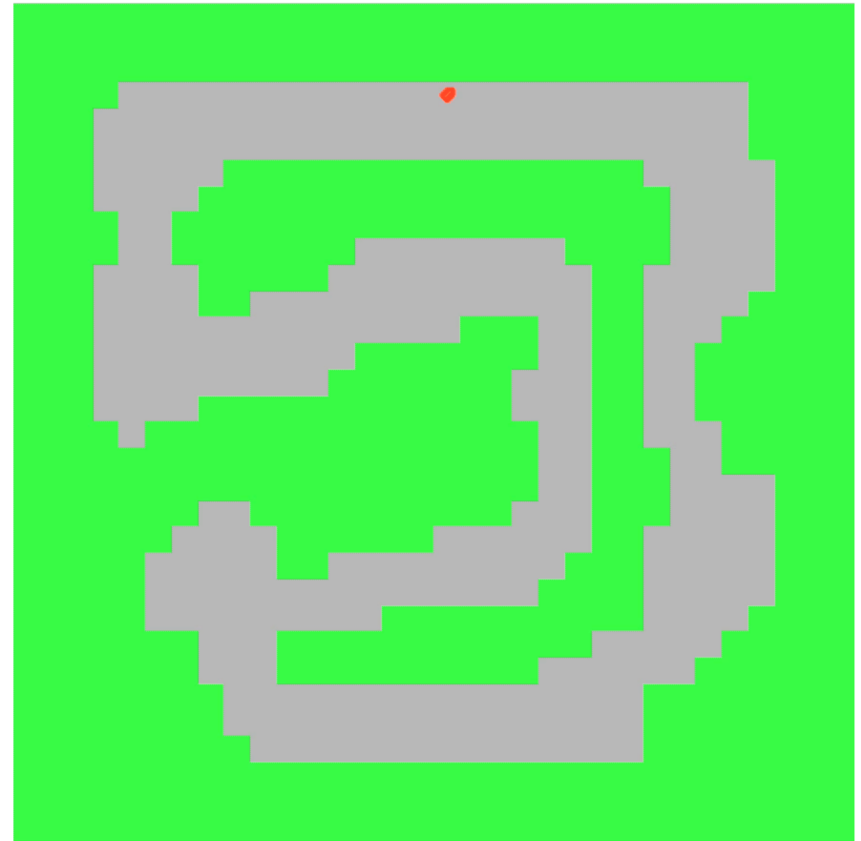
par ex. →



: à droite

: -

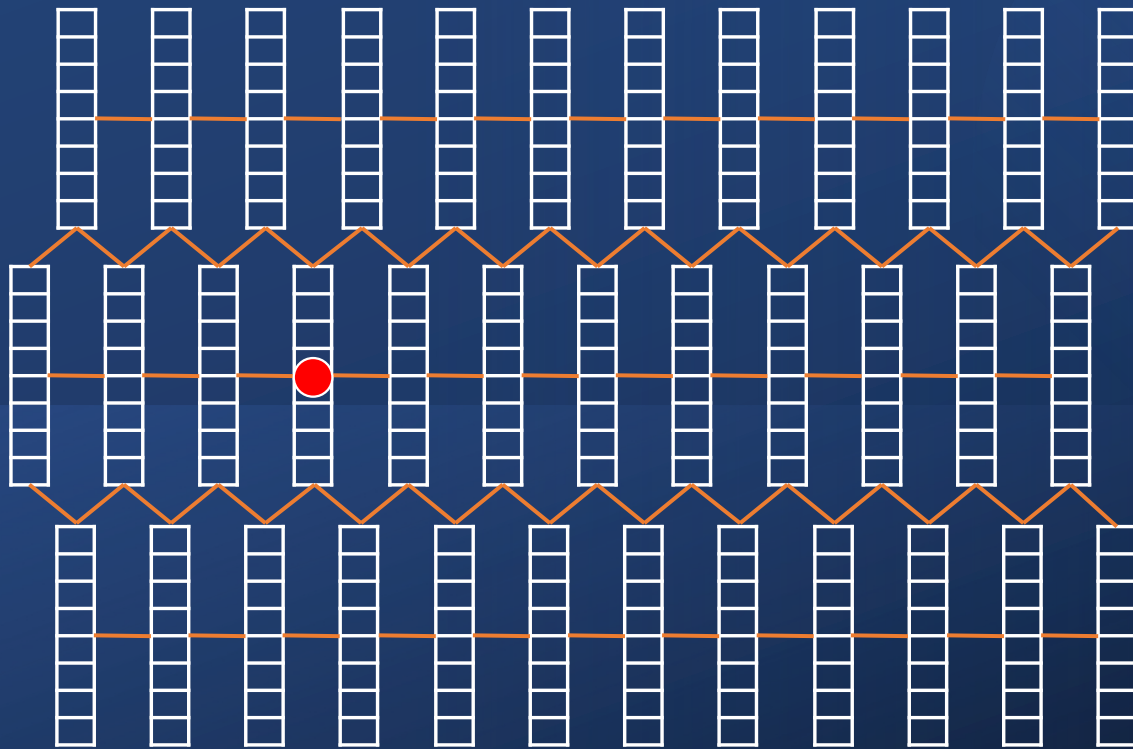
nv. situation ⇒  
nv. gène avec volant et  
accélération aléatoires





# Métaheuristique

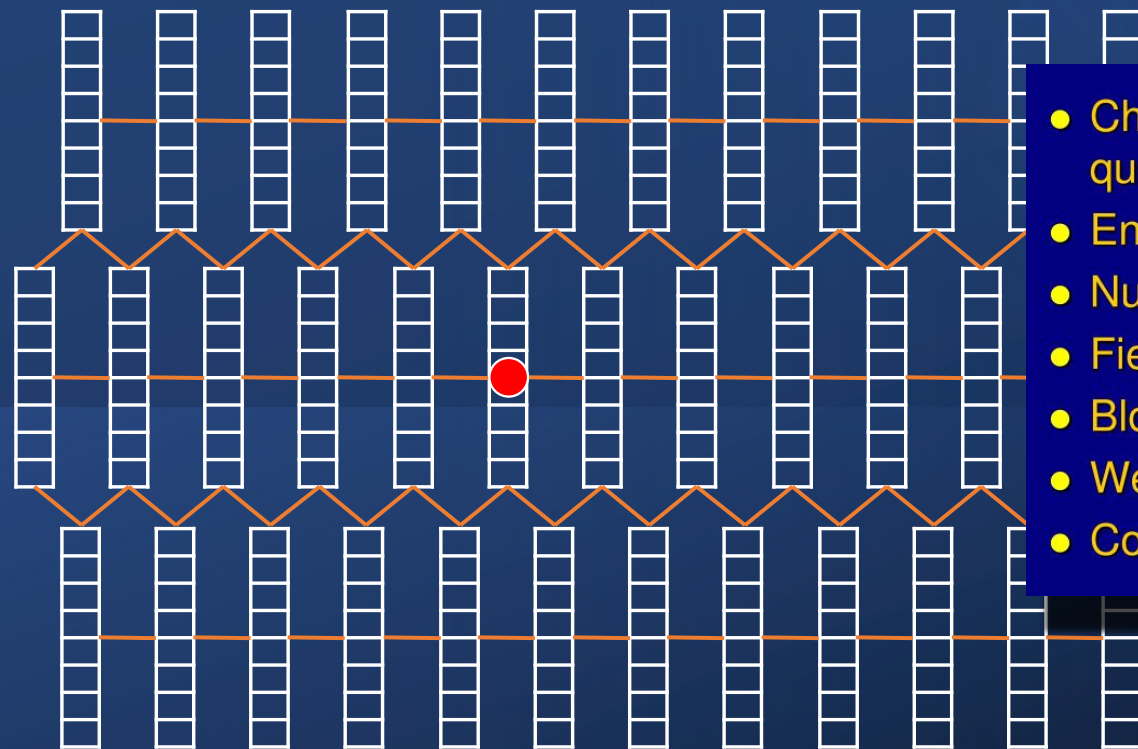
métaheuristique méthode de recherche de l'optimum dans l'espace des solutions



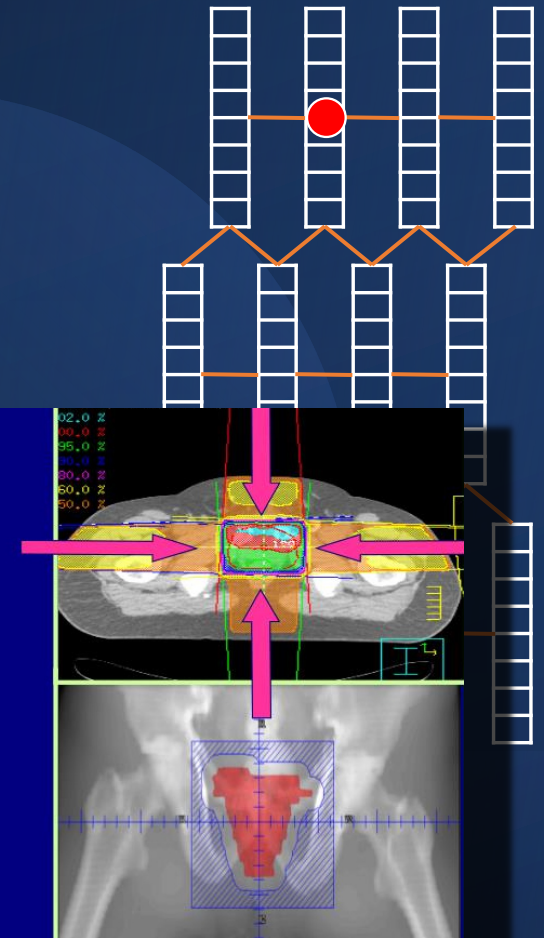


# Métaheuristique

algorithme génétique



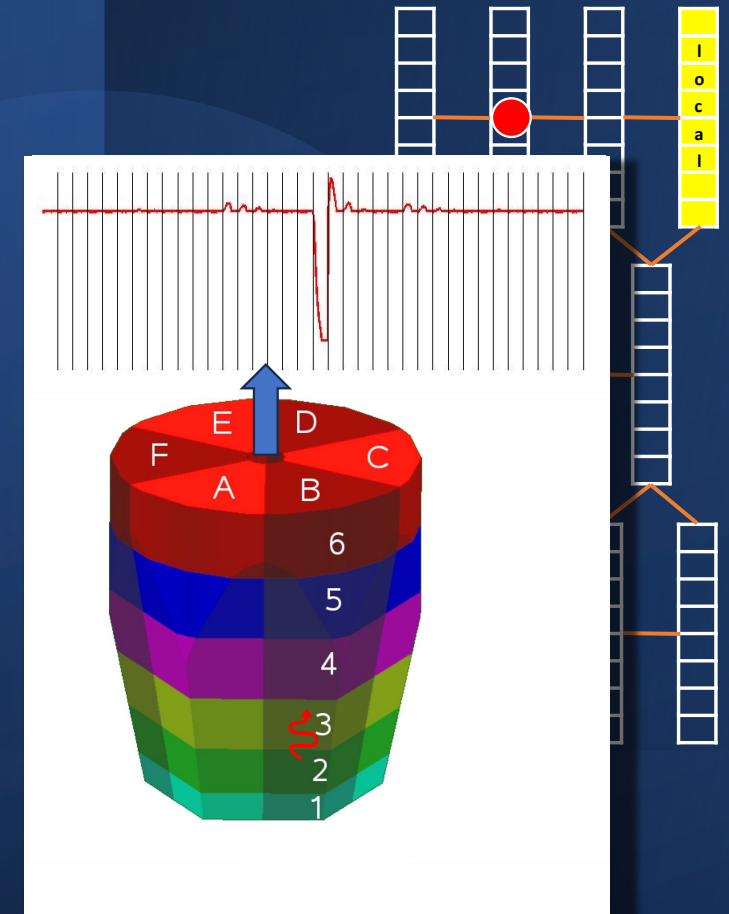
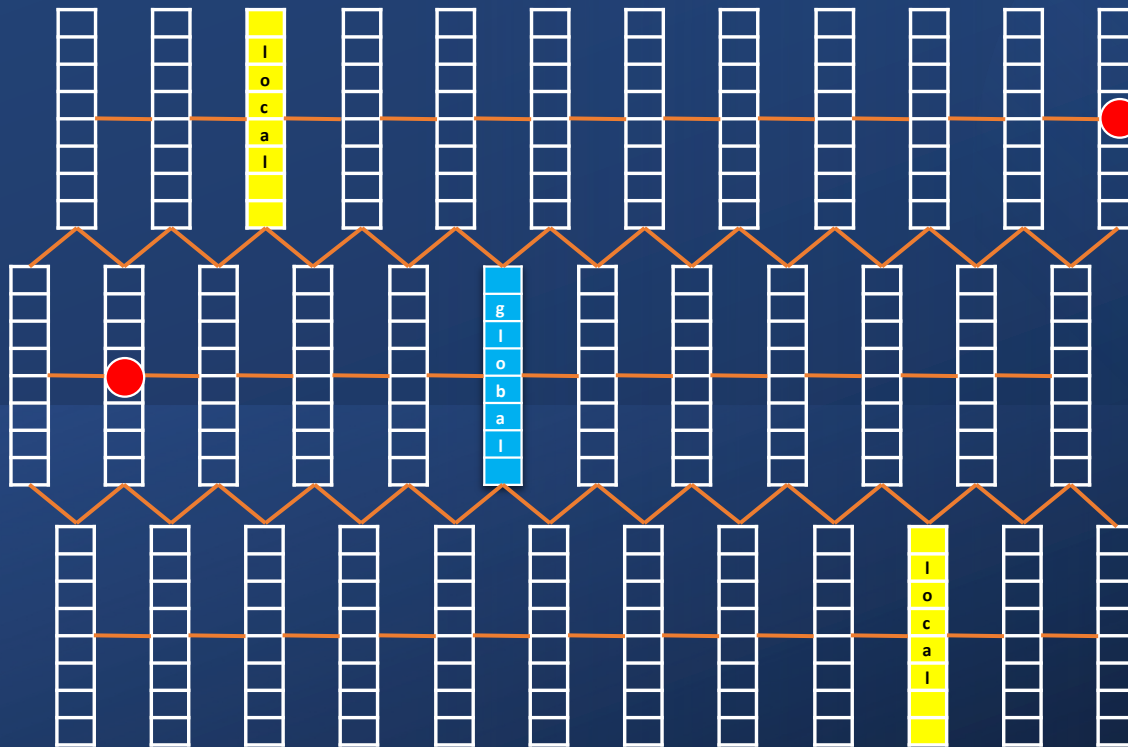
- Choice of radiation quality
- Entry point
- Number of beams
- Field size
- Blocks
- Wedges
- Compensators



Optimization of beam orientation in radiotherapy using planar geometry  
O Haas et al., *Phys. Med. Biol.* 43 2179

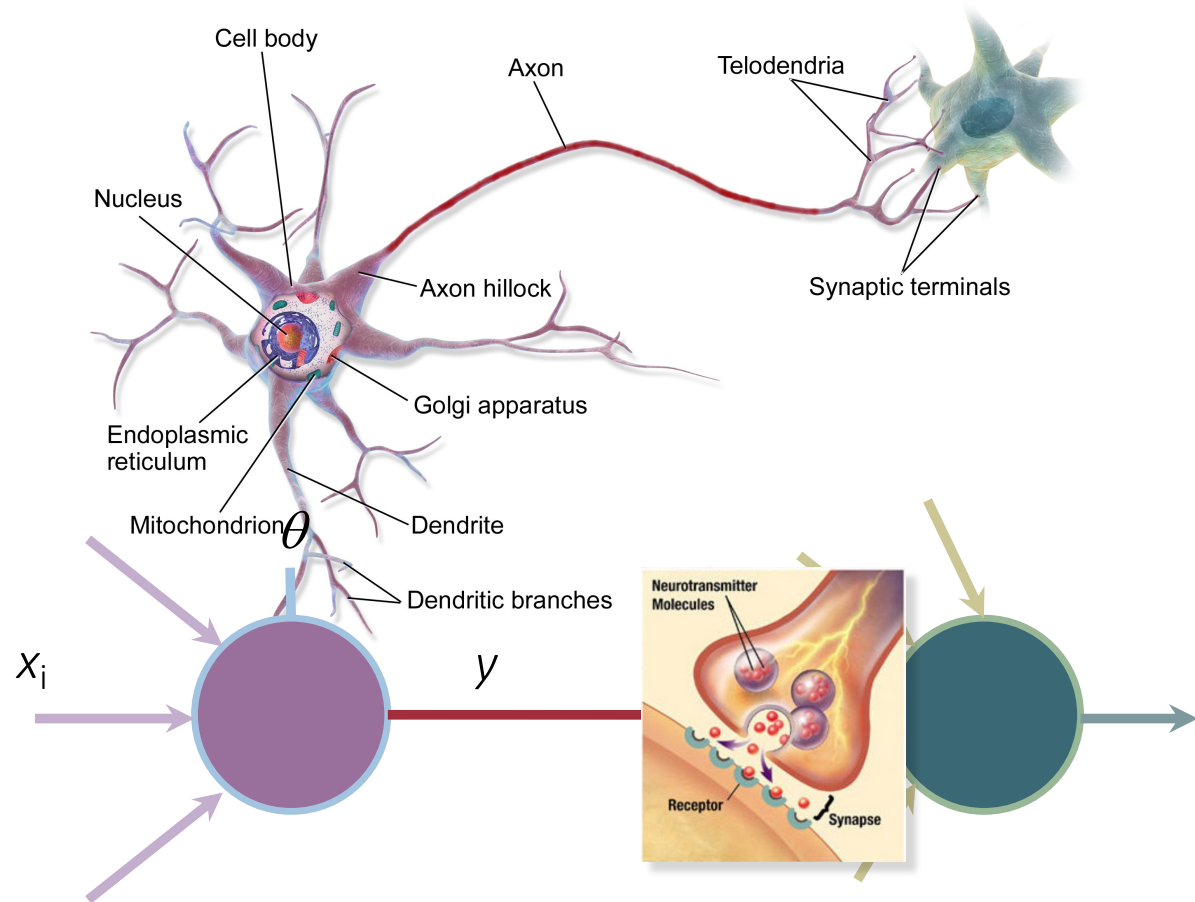
# Métaheuristique

algorithme par essaim



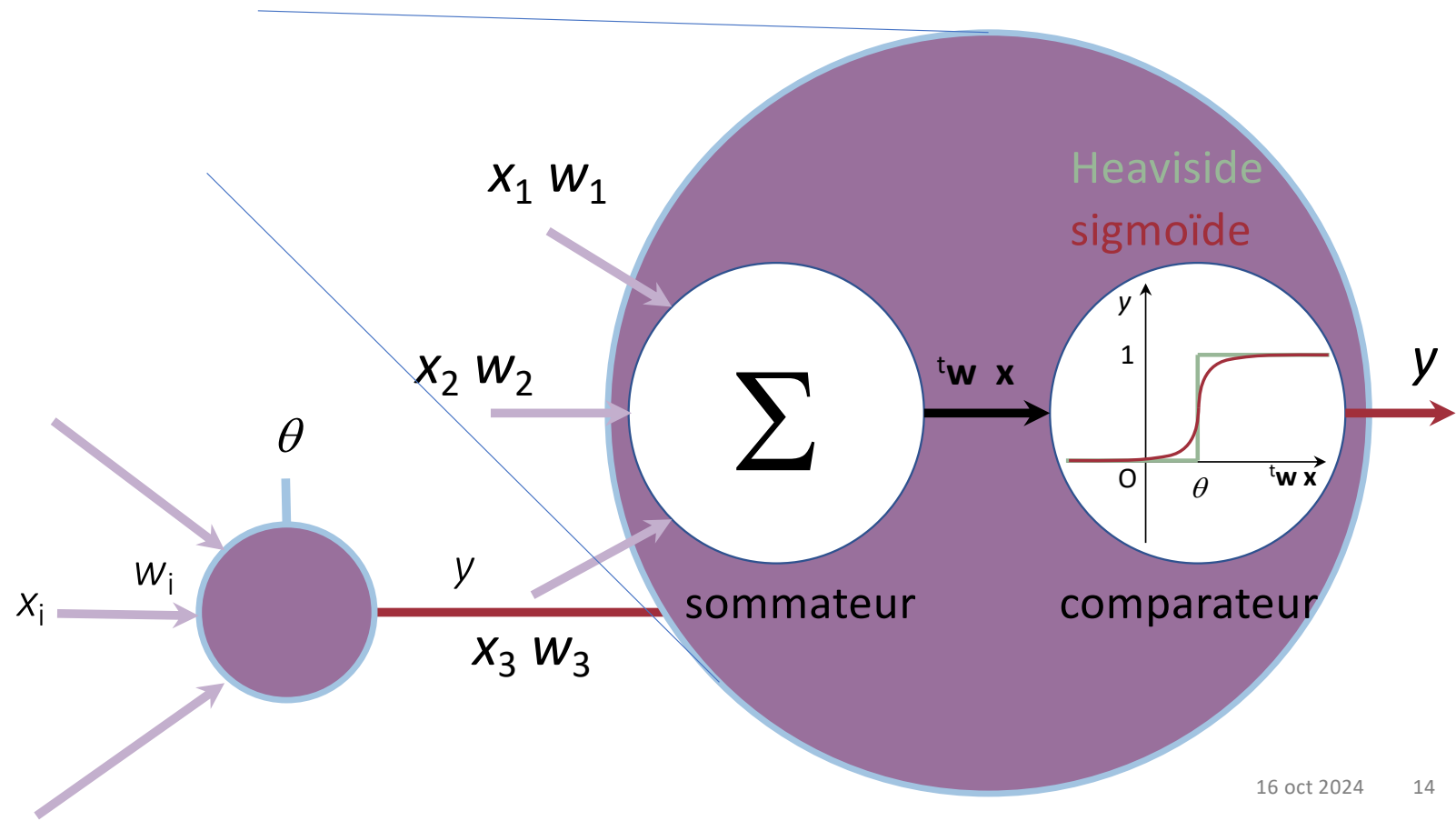
Fully informed particle swarm algorithm  
applied to AGATA  
*M. Schlarb et al., Eur. Phys. J. A (2011) 47: 131*

# Le neurone



# Le neurone

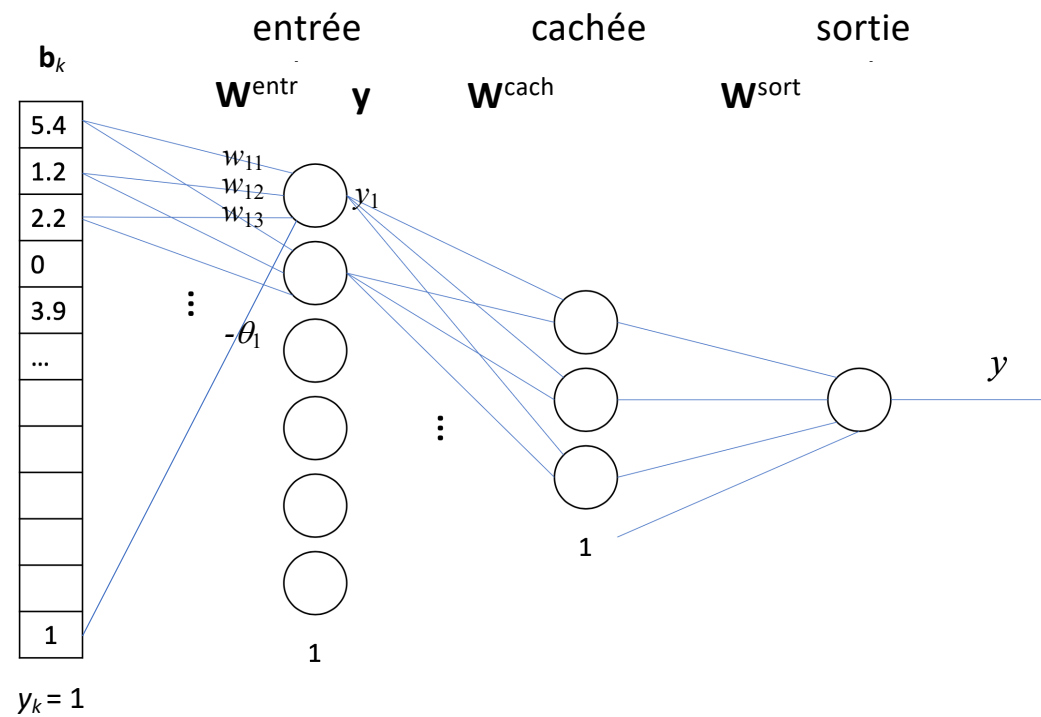
entrées   poids   biais   sortie



## réseau de neurones

neurone linéaire :  $y = \mathbf{w} \mathbf{x}$ 

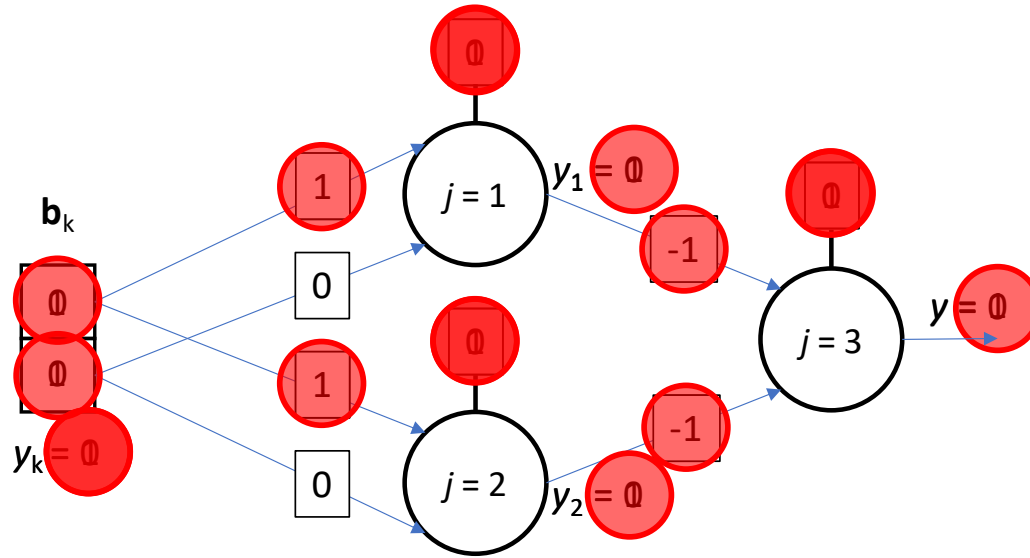
$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ 1 \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \dots \\ -\theta \end{pmatrix}$$



$$\mathbf{y} = \mathbf{W}^{\text{entr}} \mathbf{b}_k$$

$$\mathbf{y} = \mathbf{W}^{\text{sort}} \mathbf{W}^{\text{cach}} \mathbf{W}^{\text{entr}} \mathbf{b}_k$$

# réseau de neurones, optimisation



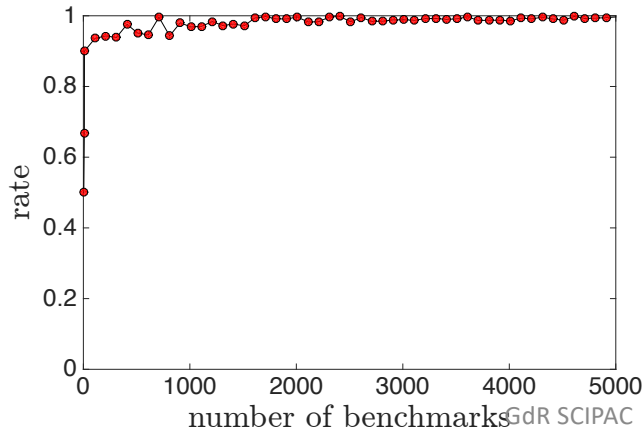
$y_k = 1$  si  $\mathbf{b}_k$  contient 1

●  $y = y_k$  ne rien faire

●  $y \neq y_k$  les paramètres ayant contribué à l'échec sont modifiés :

$y_k = 0$  : rendre plus difficile l'allumage des  $y_j = 1$   
 $\theta_j \rightarrow \theta_j + 1$   
 if  $x_{ij} = 1, w_{ij} \rightarrow w_{ij} - 1$

●  $y_k = 1$  : rendre plus facile l'allumage des  $y_j = 0$   
 $\theta_j \rightarrow \theta_j - 1$   
 if  $x_{ij} = 1, w_{ij} \rightarrow w_{ij} + 1$



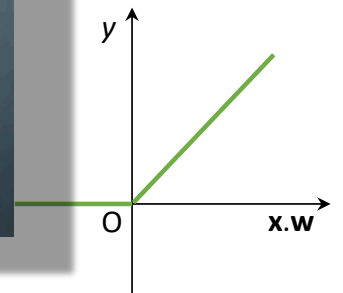
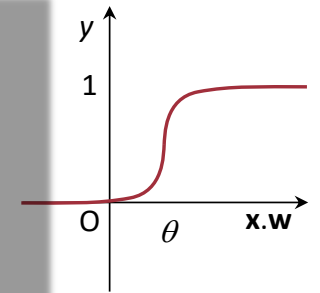
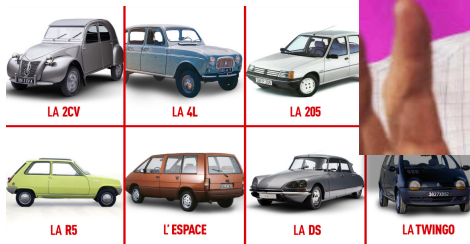


# Apprentissage profond

avant 2012



après 2012

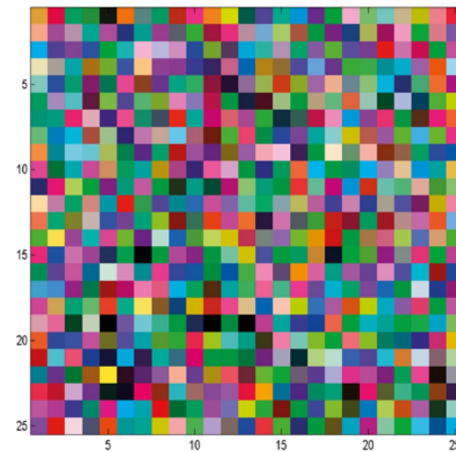


auto

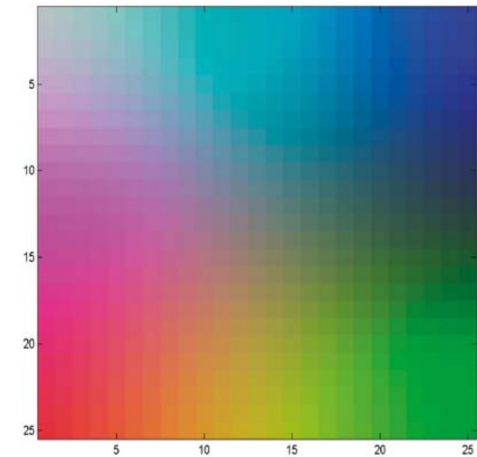
# Carte auto-adaptative (Kohonen)

## Exemple

	C	M	Y
X	.02	.99	.01
	.33	.98	.01
	.67	.92	.58
	.82	.10	.03
	.94	.33	.27
	.74	.92	.21



initialisation aléatoire



carte finale

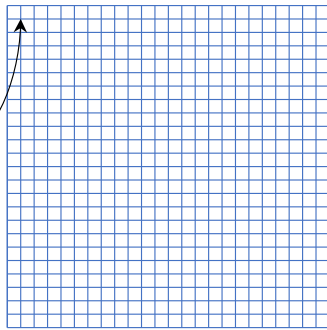
but : créer une carte /  
lignes se ressemblant sont proches



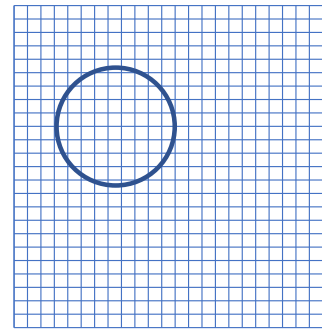
# Carte auto-adaptative (Kohonen)

## 1) Initialisation

$\forall$  nœud  $k$ :  
 $\mathbf{x}_k = (x_1, x_2, x_3, x_4, x_5)$   
 aléatoire



## 2.2) Modification du voisinage



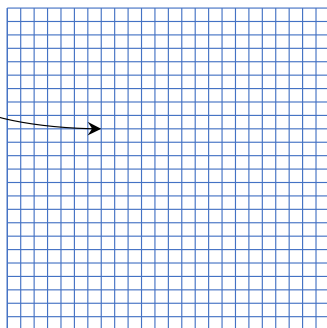
$\forall$  nœud  $k$  dans cercle:  
 $\mathbf{x}_{kj} = \mathbf{x}_{kj} + \varepsilon(r) (\mathbf{x}_{ij} - \mathbf{x}_{kj})$

## 2) Apprentissage

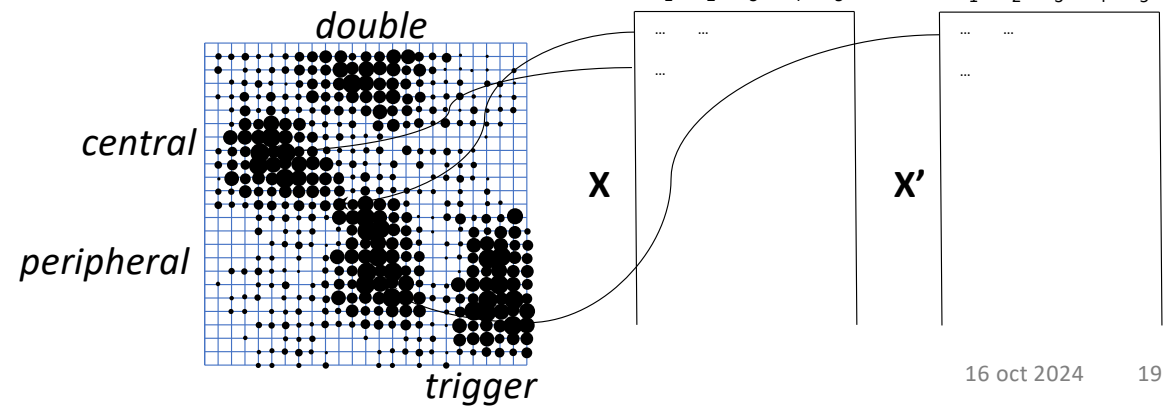
### 2.1) best match ( $d = \|\mathbf{x}_i - \mathbf{x}_k\| \min$ )

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$i$	10.1	5.21	23.1	12.2	100
	14.3	3.41	30.3	15.5	99.1
	19.2	2.65	25.4	17.7	89.2
	18.8	6.23	34.2	15.5	110
	12.7	4.03	45.3	13.3	118
	13.4	5.05	34.6	12.0	111
	12.3	6.43	35.4	18.2	103
	...	...	...	...	...

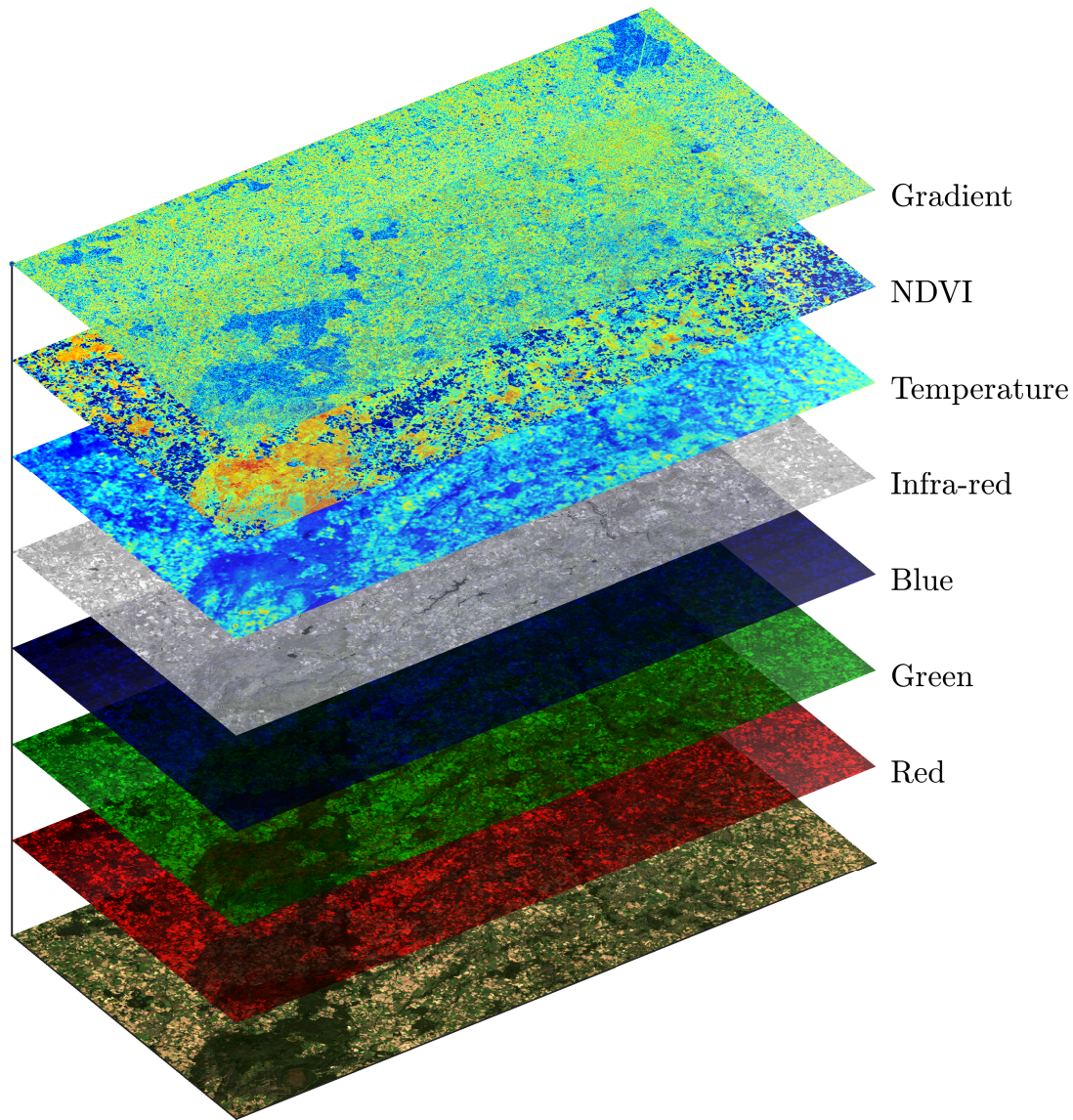
$d_{\min}$



### 3) Exploitation

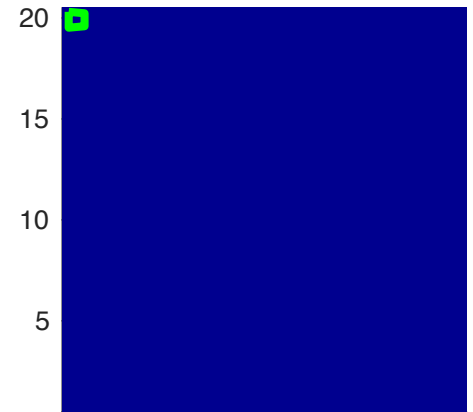
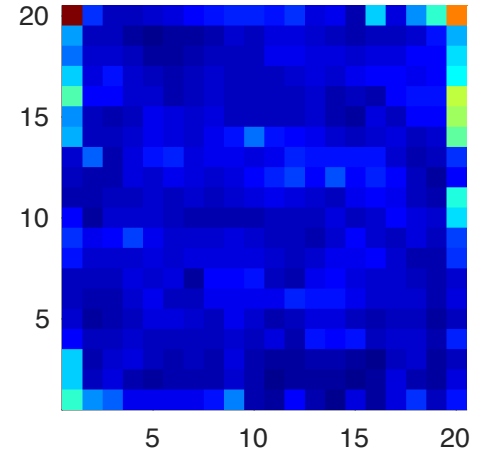
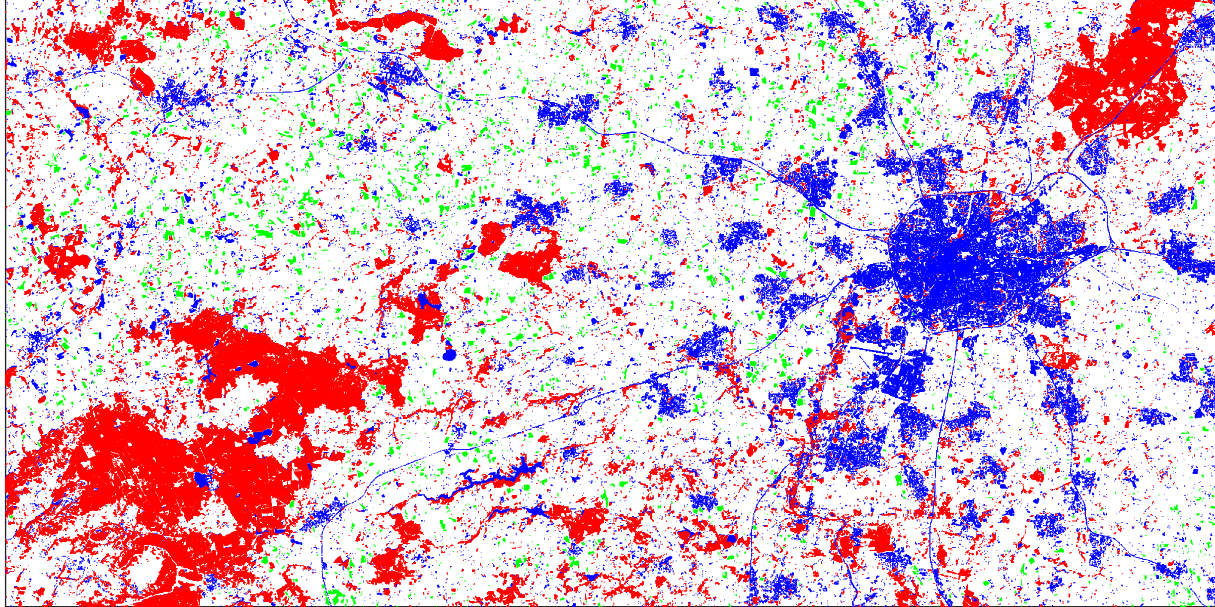
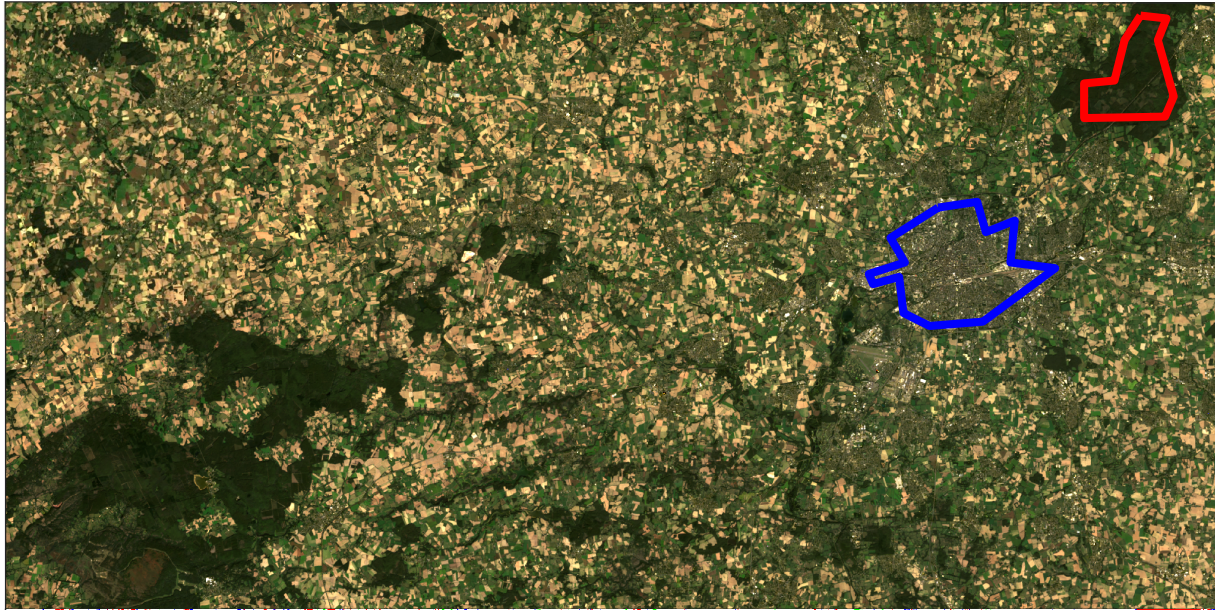


# carte auto adaptative : SIG

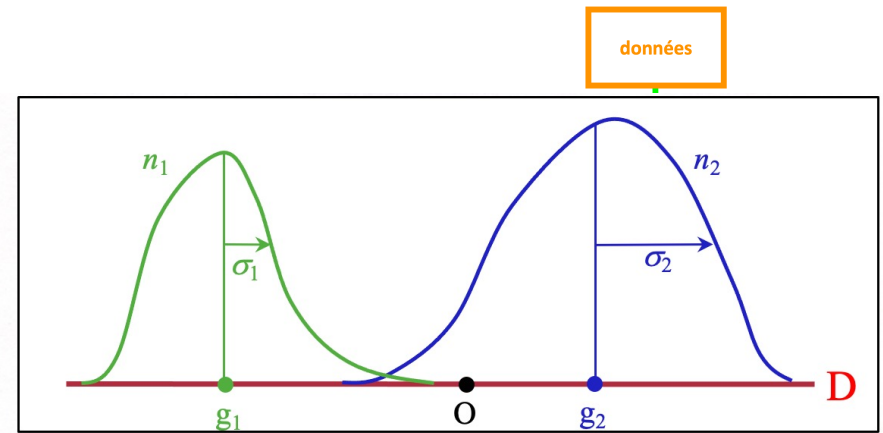
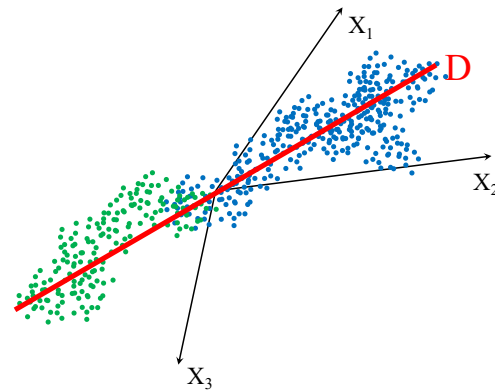




# carte auto adaptative : SIG



# Analyse Factorielle Discriminante



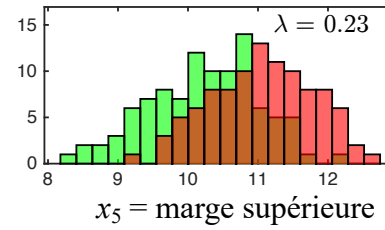
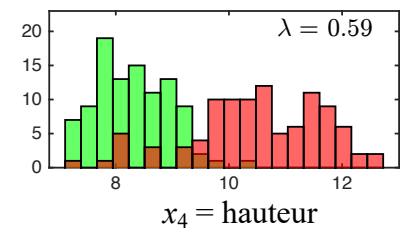
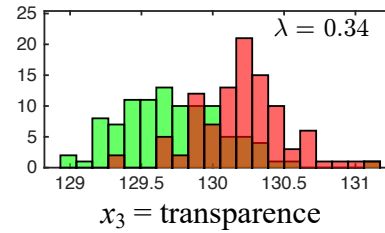
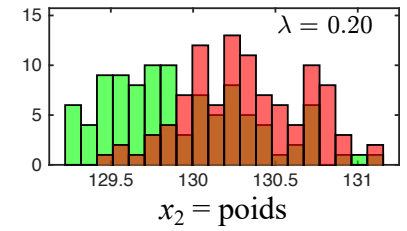
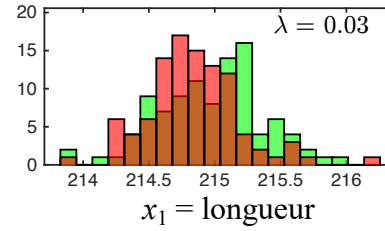
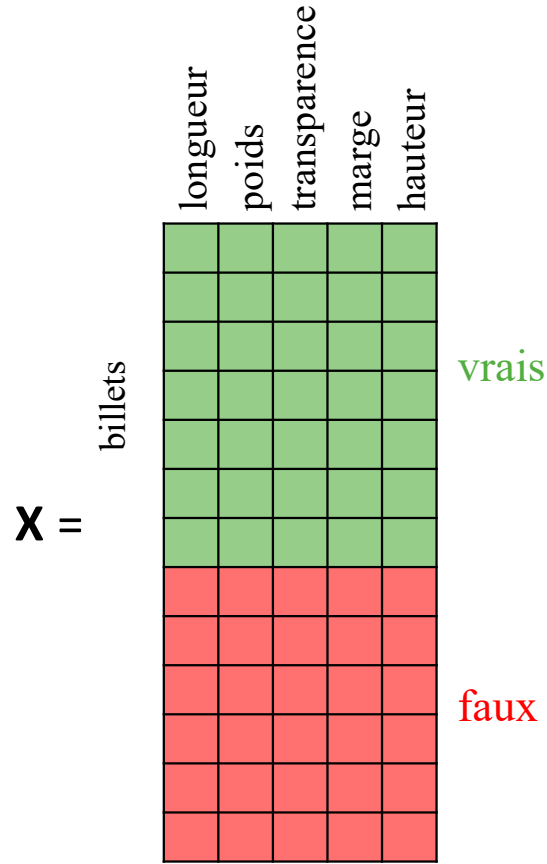
$$I^{\text{inter}} = \frac{n_1}{n} g_1^2 + \frac{n_2}{n} g_2^2$$

$$I^{\text{intra}} = \frac{n_1}{n} \sigma_1^2 + \frac{n_2}{n} \sigma_2^2$$

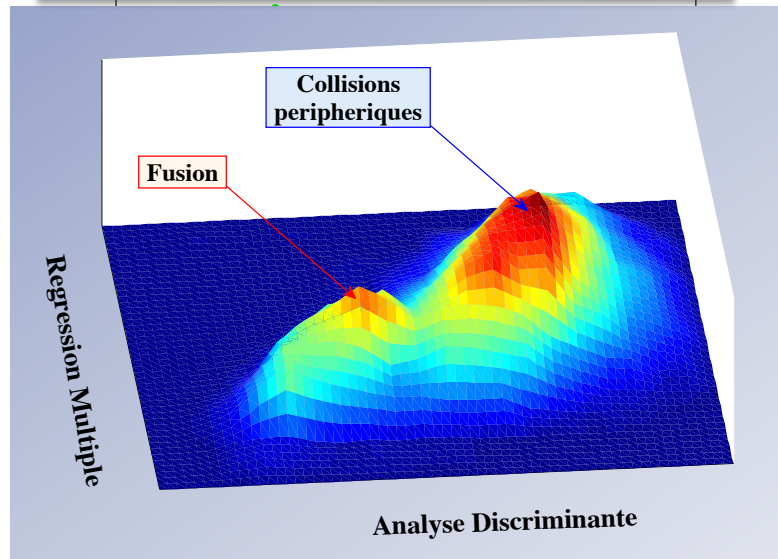
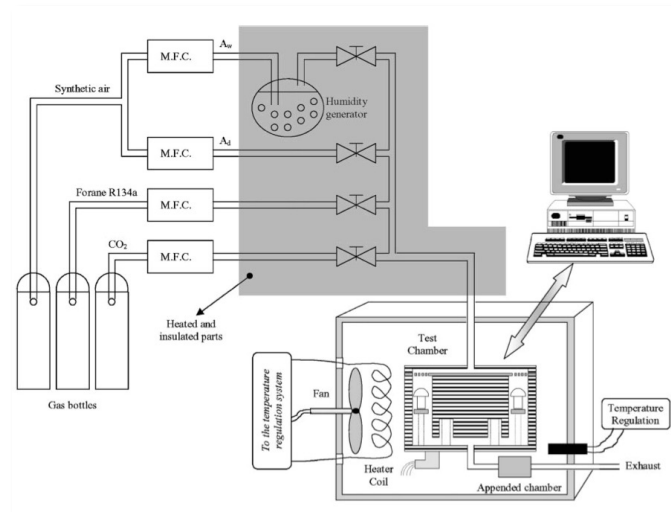
$$I = I^{\text{inter}} + I^{\text{intra}}$$

$$\lambda = \frac{I^{\text{inter}}}{I}$$

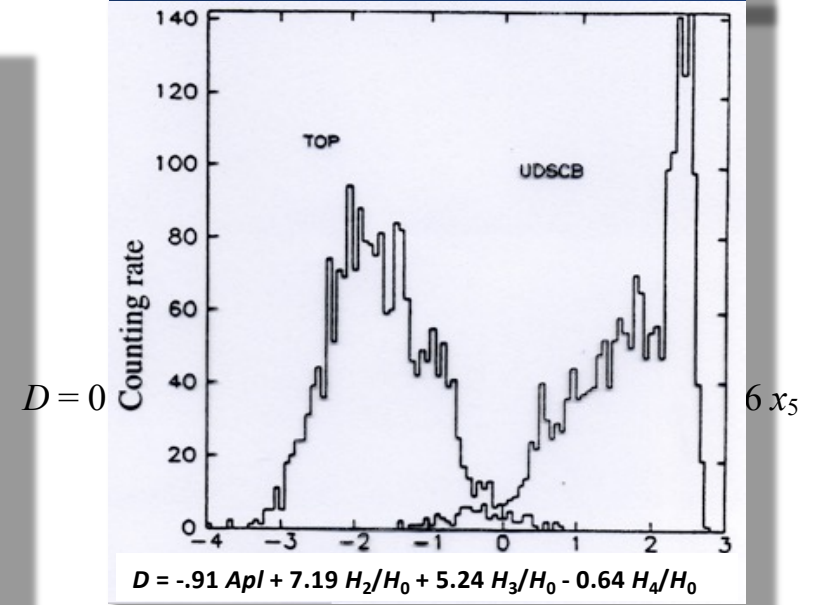
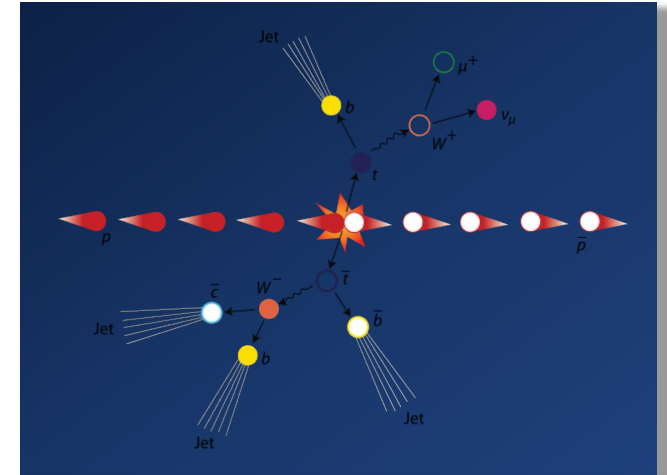
# AFD : faux billets



# AFD : faux billets



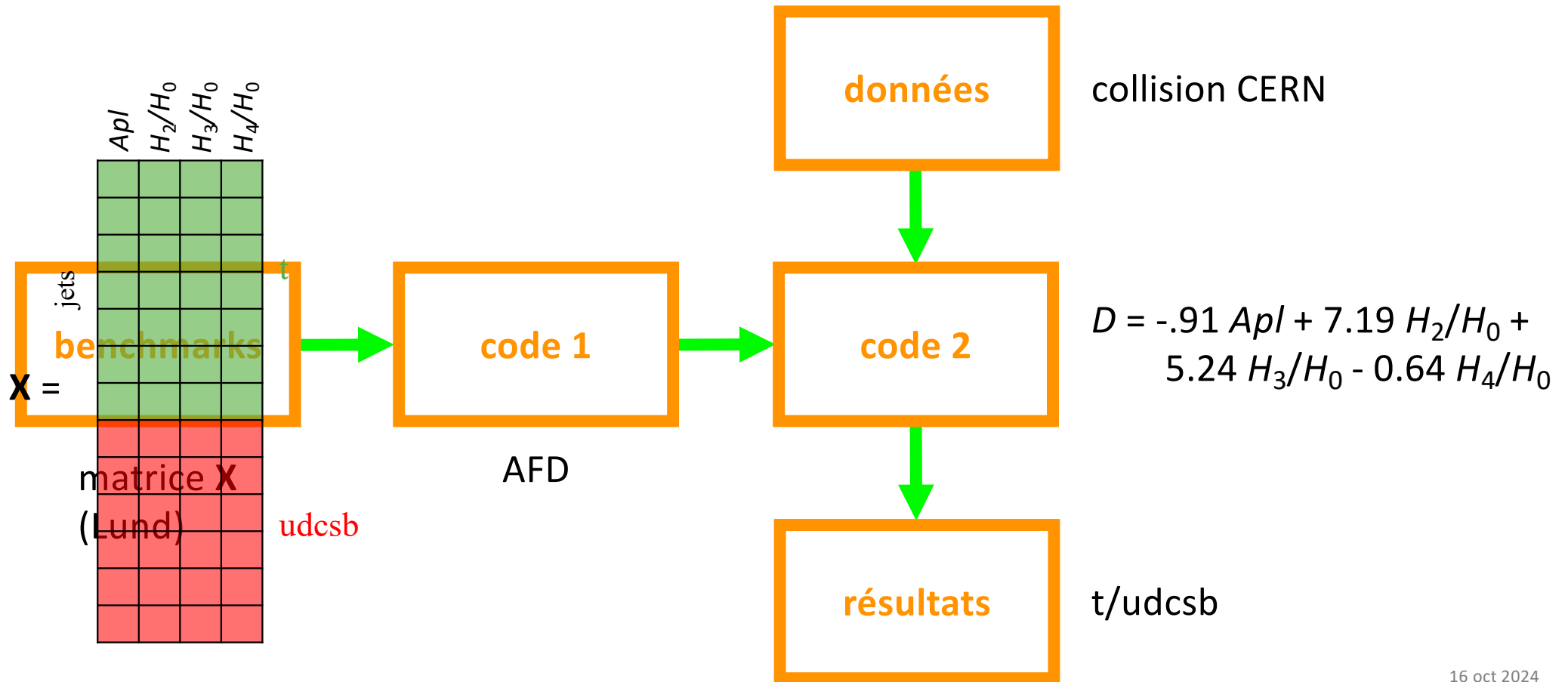
P. Désesquelles *et al.*, *Phys Rev C* **62**, 024164, 2000







M. Mjahed, thèse, Univ. Clermont II, 1987



# Analyse Factorielle Discriminante



# Substituts aux Réseaux de Neurones

- Discrimination  AFD : Analyse Factorielle Discriminante
- Classification non supervisée  Algorithme de Ward, k-moyennes
- Interpolation  Splines, Polynomes
- Régression  ACP, Régression multiple



