SYS843

D. Méta heuristique et optimisation évolutionnaire

Partie 2: Optimisation par essaims particulaires

Eric Granger

Ismail Ben Ayed

Hiver 2017

CONTENU DU COURS

A — RÉSEAUX DE NEURONES ARTIFICIELS

- introduction
- classificateurs de type perceptron monocouche
- réseaux de neurones multicouches sans rétroaction
- réseaux à base de noyaux
- Réseaux à apprentissage profond

B — SYSTÈMES FLOUS

- sous-ensembles et opérations flous
- variables linguistiques et règles d'inférence
- les α-coupes et normes triangulaires
- logique floue pour la prédiction, monitoring et contrôle

C — ALGORITHMES DE CATÉGORISATION

- réseaux de neurones auto-organisateurs
- algorithme fuzzy C-means
- catégorisation à noyaux et spectrale

D — MÉTA HEURISTIQUE ET OPTIMISATION ÉVOLUTIONNAIRE

- algorithmes génétiques
- optimisation par essaims particulaires
- optimisation avec multiples critères
- applications aux systèmes neuronales et flous

CONTENU DU COURS

D.2 Optimisation par essaims particulaires

1) **Intelligence d'essaims**
2) Algorithme PSO canonique
3) Variantes de PSO
   - Environnements dynamiques
   - Optimisation multicritère
4) Application – Optimisation évolutionnaire de RNA

1) Intelligence par essaims

Définitions

- **Une famille de techniques en AI:**
  – Les systèmes sont typiquement constitués d'une population d'agents simples qui interagissent entre eux, et avec leur environnement
  – Aucun contrôle centralisé, mais l'interaction entre les agents permet un comportement global
  – Inspiré des phénomènes naturels: colonies fourmis, poissons, flocage d'oiseaux, etc.

1) Intelligence par essaims

Exemple d'application

- **Robotique distribuée:**

  Swarm-bots: Swarms of self-assembling artifacts

  – Une population de robots mobiles (*s-bots*) autonomes qui peuvent s'auto-organiser pour naviguer, percevoir et manipuler

  Swarmanoid: Towards Humanoid Robotic Swarms

  – Systèmes de robotique distribués, conçus avec des petits robots autonomes qui sont interconnectés dynamiquement

  – https://youtu.be/Hyk3D6j1DsU

1) Intelligence par essaims

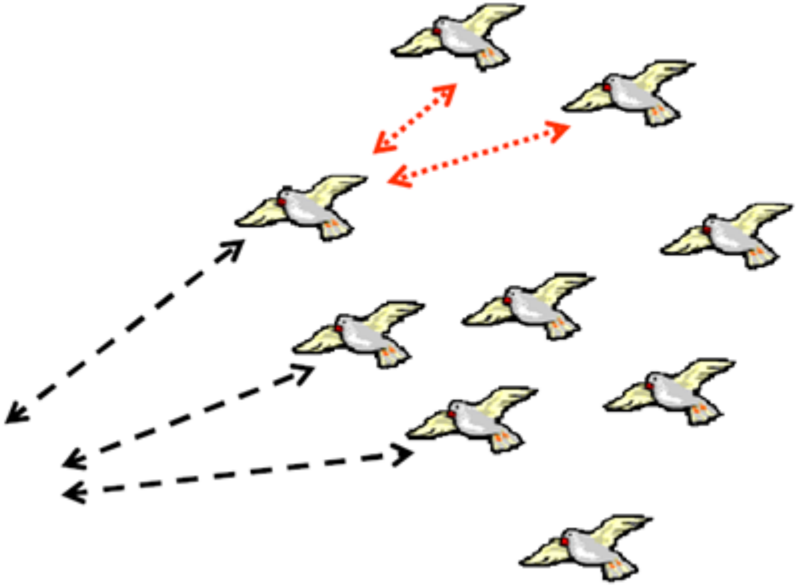
Définitions

- **Optimisation par essaims particulaires (ou *Particle Swarm Optimization*, PSO):**
  - Technique d'optimisation stochastique à base de populations
  - Développé par by Eberhart et Kennedy en 1995
  - Le PSO est initialisé avec une population aléatoire de solutions potentielles (essaim de particules) dans l'espace de recherche, et explore l'espace pour un optima global
  - Les particules planent sur l'espace de recherche, guidées selon l'emplacement des particules avec la meilleure fitness

1) Intelligence par essaims

Concept général

- **Un algorithme PSO imite le comportement social d'animaux et d'insectes**
  - Les membres de la population interagissent entre eux tout en apprenant de leur propre expérience
  - Ils bougent graduellement dans des meilleures régions de l'espace des solutions



1) Intelligence par essaims

Concept général

- **PSO fait évoluer un essaim de particules:**
  - chacun réside à un endroit dans l'espace de recherche
  - le coût (ou *fitness*) lié avec chaque particule indique la qualité de sa position dans l'espace
- **Les particules planent sur l'espace de recherche avec une certaine vélocité (direction et vitesse) qui est influencée par:**
  1. la meilleure position qu'elle a trouvé à date, et
  2. la meilleure solution trouvé à date par ses voisins
- **L'essaim converge éventuellement à des positions 'optimales'**

1) Intelligence par essaims

Concept général

- **La vélocité des particules (direction et vitesse) est guidée par deux composantes:**
  1. cognitives – son expérience antérieure d'exploration, et
  2. sociales – l'expérience d'exploration dans son voisinage
- **Avantages:**
  - peut converger rapidement vers des bonnes solutions
  - implémentations simples, avec peu de paramètres
  - versatilité: peut résoudre beaucoup de différents problèmes
- **Applications aux problèmes avec:**
  - un espace de recherche continu, discret ou mixte
  - optimisation dynamique et multicritère, avec 1+ minimums locaux

1) Intelligence par essaims

PSO versus AG

- **Lien avec les techniques de calcul évolutionnaires (e.g., algorithmes génétiques, AG)**
  - Comme un AG, le PSO exploite aussi des populations de solutions pour chercher l'optima, mais…
  - PSO ne suit aucun concept de 'survie des plus forts', même s'il exploite le concept de *fitness*
  - PSO n'utilise pas d'opérateurs évolutionnaires comme celles de mutation et de croisement
  - chaque particule évolue selon son expérience antérieure, et selon ses relations avec les autres particules de l'essaim

…par contre, certains algorithmes hybrides intègrent des concepts d'AG et de PSO

CONTENU DU COURS

D.2 Optimisation par essaims particulaires

1) Intelligence d'essaims
2) **Algorithme PSO canonique**
3) Variantes de PSO
   - Environnements dynamiques
   - Optimisation multicritère
4) Application – Optimisation évolutionnaire de RNA

2) Algorithme PSO canonique

Particle Swarm Optimization

LIACS Natural Computing Group Leiden University

basé sur l'article

R. Poli, J. Kennedy, T. Blackwell, 'Particle Swarm Optimization: An Overview,' *Swarm Intelligence*, **1:1**, 33-57, 2007.

2) Algorithme PSO canonique

Pseudo-code of a basic PSO

```
Randomly generate an initial population
repeat
    for i = 1 to population_size do
        if f(x_i) < f(p_i) then p_i = x_i;
        p_g = min(p_neighbours);
        for d =1 to dimensions do
            velocity_update();
            position_update();
        end
    end
until termination criterion is met.
```

2) Algorithme PSO canonique

Algorithme

1. **Randomly initialize particle positions and velocities**

2. **While not terminated**
   - For each particle i:
     - Evaluate fitness y_i at current position $\mathbf{x}_i$
     - If y_i is better than $pbest_i$ then update $pbest_i$ and $\mathbf{p}_i$
     - If y_i is better than $gbest_i$ then update $gbest_i$ and $\mathbf{g}_i$
   - **For each particle**
     - Update velocity $\mathbf{v}_i$ and position $\mathbf{x}_i$ using:

$$\begin{cases} \vec{v}_i \leftarrow \vec{v}_i + \vec{U}(0,\varphi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0,\varphi_2) \otimes (\vec{g}_i - \vec{x}_i) \\ \vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i \end{cases}$$

2) Algorithme PSO canonique

Notation

For each particle *i* :

- $\mathbf{x}_i$: a vector denoting its position
- $\mathbf{v}_i$: the vector denoting its velocity
- y_i : denotes the fitness score of $\mathbf{x}_i$
- $\mathbf{p}_i$: the best position that it has found so far
- $\boldsymbol{pbest}_i$: denotes the fitness of $\mathbf{p}_i$
- $\mathbf{g}_i$: the best position that has been found so far in its neighborhood
- $\boldsymbol{gbest}_i$: denotes the fitness of $\mathbf{g}_i$

2) Algorithme PSO canonique

Notation

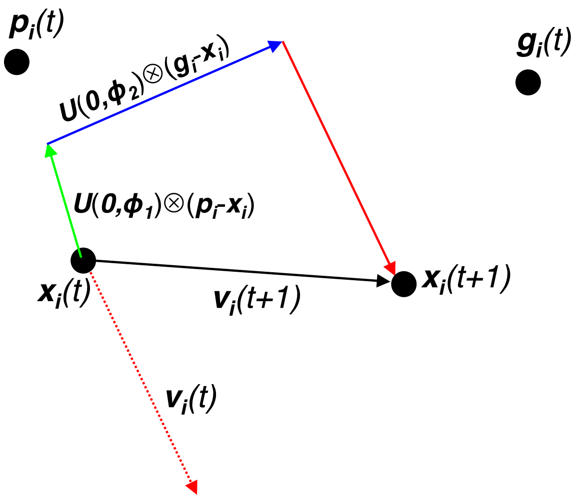
Velocity update:

- **U**(0, ϕ_i) : a random vector uniformly distributed in [0, ϕ_i] regenerated at every generation for each particle
- ϕ_1 : the acceleration coefficients determining the scale of the force in direction of $\mathbf{p}_i$
- ϕ_2 : are the acceleration coefficients determining the scale of the force in direction of $\mathbf{g}_i$
- $\otimes$: denotes the element-wise multiplication operator

2) Algorithme PSO canonique

Velocity Update

$$\vec{v}_i \leftarrow \vec{v}_i + \vec{U}(0,\varphi_1)\otimes(\vec{p}_i - \vec{x}_i) + \vec{U}(0,\varphi_2)\otimes(\vec{g}_i - \vec{x}_i)$$



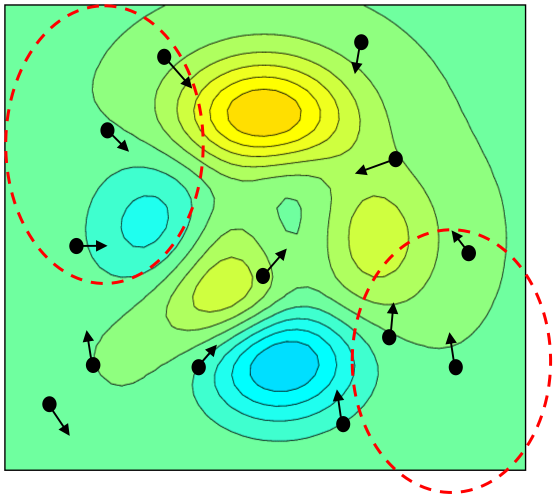
- **Momentum**: The force pulling the particle to continue its current direction
- **Cognitive component**: The force emerging from the tendency to return to its own best solution found so far
- **Social component**: The force emerging from the attraction of the best solution found so far in its neighborhood

2) Algorithme PSO canonique

Neighborhood Topologies

For the social component, the neighborhood of each particle is defined by its communication structure (its social network):

1. **Geographical neighborhood topologies:**
   - Based on Euclidean proximity in the search space
   - Close to the real-world paradigm but computationally expensive

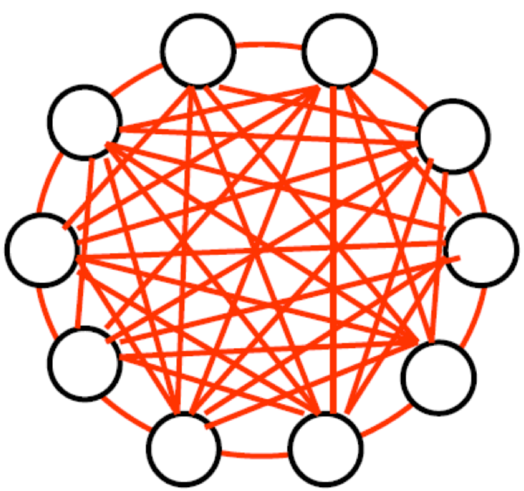


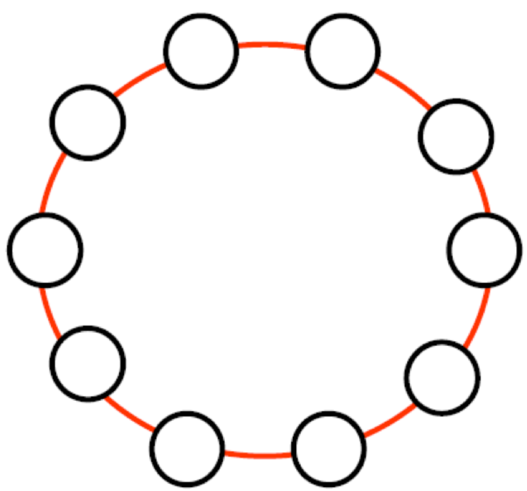
2) Algorithme PSO canonique

Neighborhood Topologies

2. Communication network topologies:

- Communication networks based on some connection graph architecture (e.g., rings, stars, von Neumann networks)
- Favored over geographical neighborhood because of better convergence properties and fewer computations





- **gbest:** each particle is influenced by the best found from the entire swarm
- **lbest:** each particle is influenced only by the particles in its local neighborhood

2) Algorithme PSO canonique

Neighborhood Topologies

- **There is no clear way of selecting the best topology for a given problem**
- **Compromise between exploration - exploitation**
  - some neighborhood topologies are better for local search, while others for global search

    lbest: topologies seem better for the global distributed search of an optima
    gbest topologies seem better for local search of an optima (it propagates information fastest to the entire population)

2) Algorithme PSO canonique

Synchronous *vs.* Asynchronous

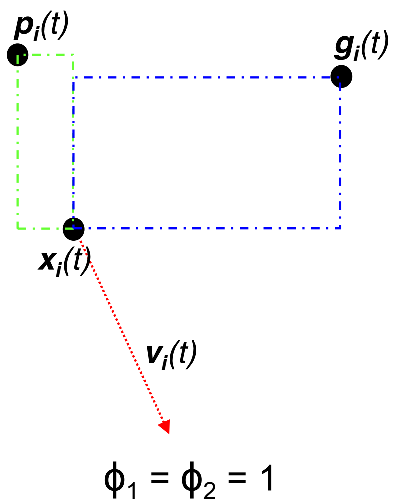
- **Synchronous updates:**
  - Personal best and neighborhood bests are updated separately from position and velocity vectors
  - Slower feedback about best positions
  - Better for **gbest** PSO

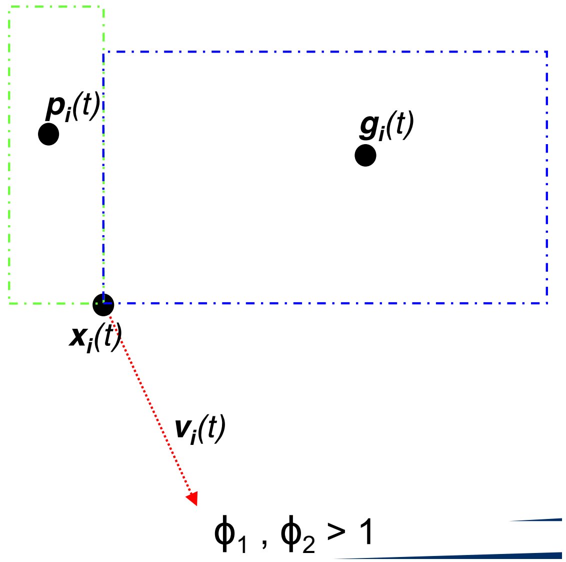
- **Asynchronous updates:**
  - New best positions updated after each particle position update
  - Immediate feedback about best regions of the search space
  - Better for **lbest** PSO

2) Algorithme PSO canonique

Acceleration Coefficients: ϕ

- The boxes show the distribution of the random vectors of the attracting forces of the **lbest** and **gbest**
- The acceleration coefficients determine the scale distribution of the random cognitive and social component vectors

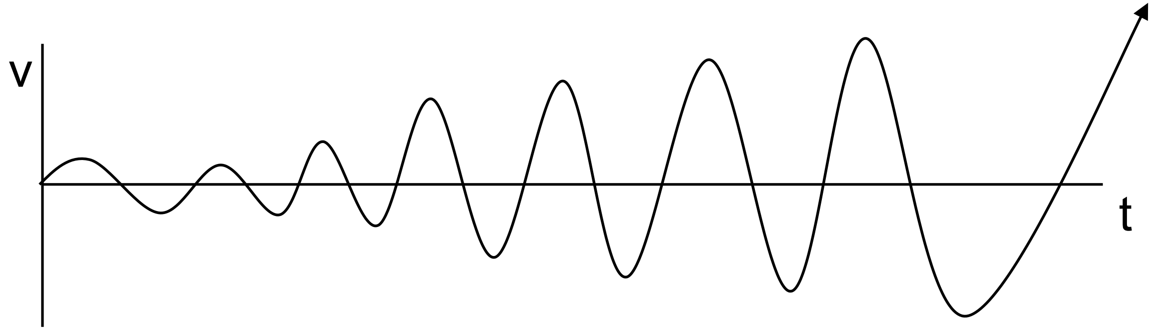




2) Algorithme PSO canonique

Stability problem

- The acceleration coefficients should be set sufficiently high, but higher acceleration coefficients result in less stable systems where velocity tends to explode.
- **Solution:** keep the velocity $\mathbf{v}_i$ within the range $[-\mathbf{v}_{max} ; +\mathbf{v}_{max}]$.



- However, limiting the velocity does not necessarily prevent particles from leaving the search space, nor does it guarantee convergence

2) Algorithme PSO canonique

Inertia weighted PSO

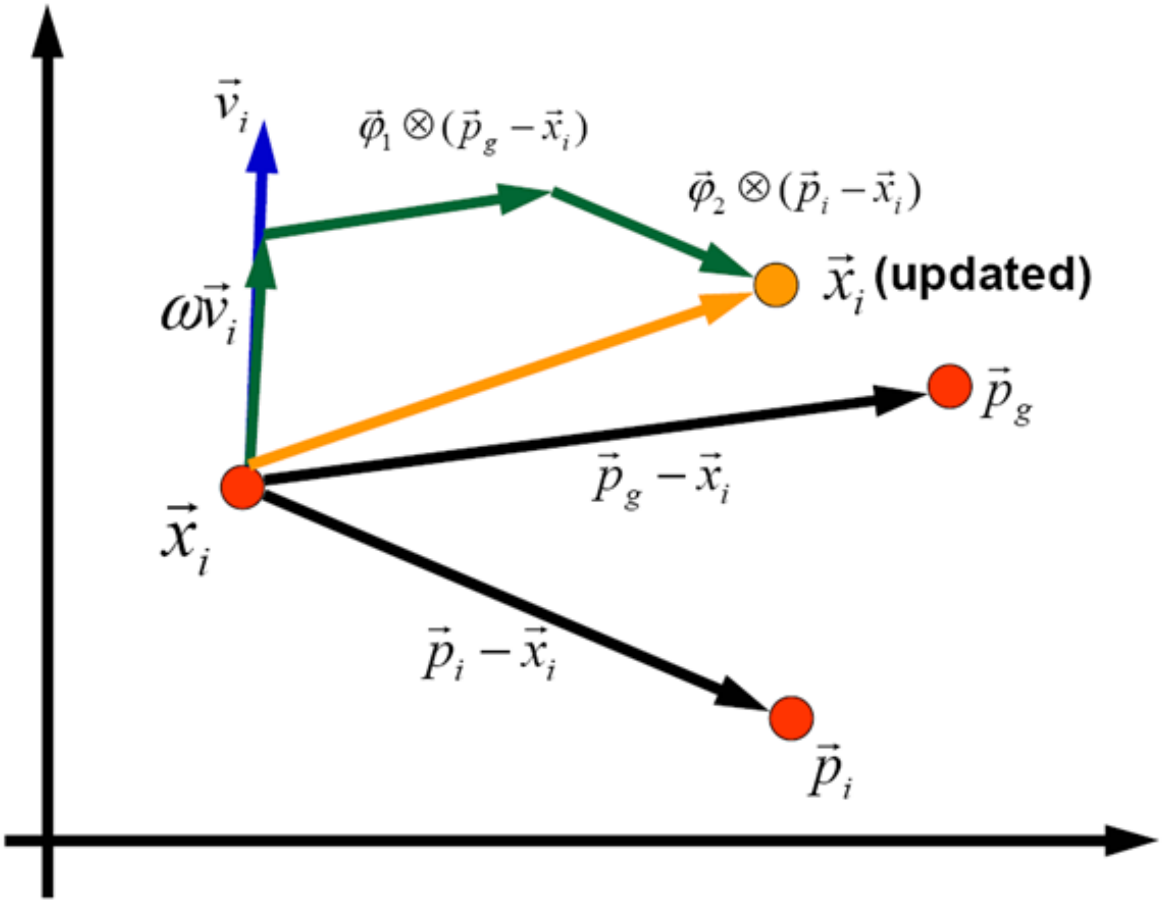
- **Solution:** an inertia weight ω to control the velocity explosion:

$$\vec{v}_i \leftarrow \omega \vec{v}_i + \vec{U}(0,\varphi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0,\varphi_2) \otimes (\vec{g}_i - \vec{x}_i)$$

- If ω, ϕ_1 and ϕ_2 are set correctly, this update rule allows for convergence without using $\mathbf{v}_{max}$
- Weight ω can be used to control the balance between exploration and exploitation:
  - if $\omega \geq 1$: velocities increase over time, swarm diverges
  - if $0 < \omega < 1$: particles decelerate, convergence depends ϕ_1 and ϕ_2
- **Rule-of-thumb:** set $\omega = 0.7298$ and $\phi_1 = \phi_2 = 1.49618$

2) Algorithme PSO canonique

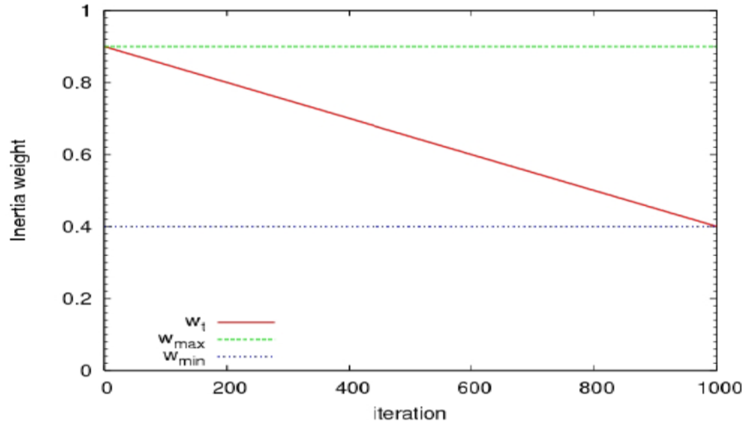
Visualizing PSO



2) Algorithme PSO canonique

Inertia weighted PSO

- **Solution:** time decreasing inertia weight
- It has been suggested to decrease ω over time (typically from 0.9 to 0.4) and thereby gradually change from an exploration to exploitation

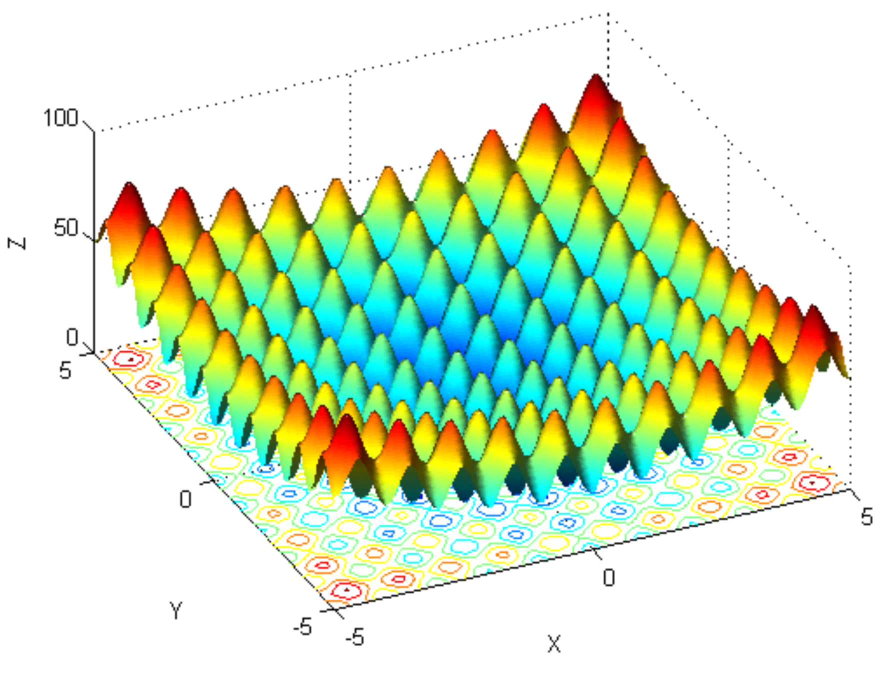


- Other schemes for a dynamically changing inertia weight have also been proposed

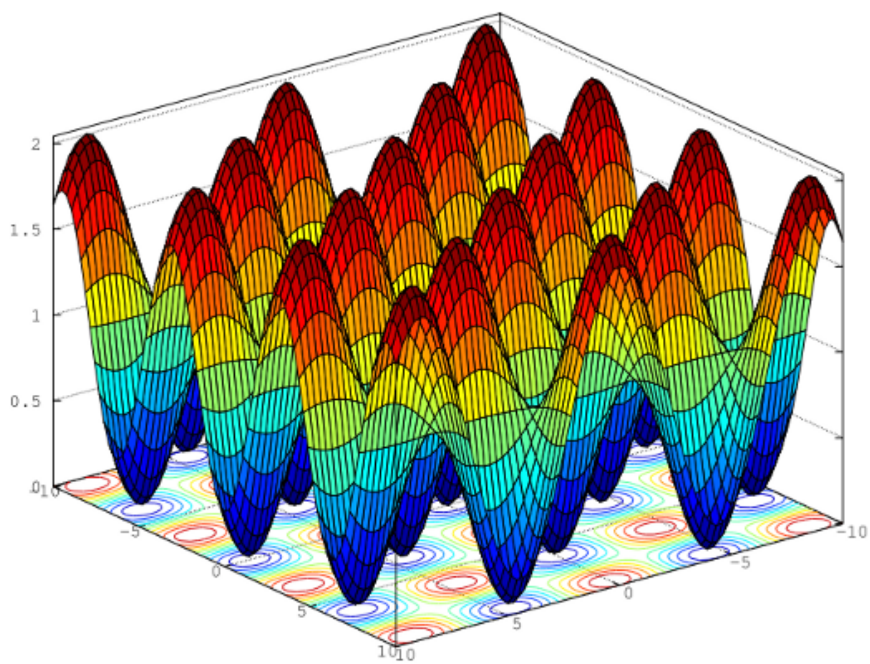
2) Algorithme PSO canonique

Examples de functions pour benchmarking

Rastrigrin



Griewank



CONTENU DU COURS

D.2 Optimisation par essaims particulaires

1) Intelligence d'essaims
2) Algorithme PSO canonique
3) **Variantes de PSO**
   - **Environnements dynamiques**
   - **Optimisation multicritère**
4) Application – Optimisation évolutionnaire de RNA

D.2(3) Variantes de PSO

Particle Swarm Optimization

And Introduction and its Recent developments

X. Li et A. P . Engelbrecht

(Tutorial prepared for GECCO 2007)

D.2(3) Variantes de PSO

Some PSO variants

Tribes (Clerc, 2006) – aims to adapt population size, so that it does not have to be set by the users; Tribes have also been used for discrete, or mixed (discrete/continuous) problems.

ARPSO (Riget and Vesterstorm, 2002) – uses a diversity measure to alternate between 2 phases;

Dissipative PSO (Xie, et al., 2002) – increasing randomness;

PSO with self-organized criticality (Lovbjerg and Krink, 2002) – aims to improve diversity;

Self-organizing Hierachicl PSO (Ratnaweera, et al. 2004);

FDR-PSO (Veeramachaneni, et al., 2003) – using nearest neighbour interactions;

PSO with mutation (Higashi and Iba, 2003; Stacey, et al., 2004)

Cooperative PSO (van den Bergh and Engelbrecht, 2005) – a cooperative approach

DEPSO (Zhang and Xie, 2003) – aims to combine DE with PSO;

CLPSO (Liang, et al., 2006) – incorporate learning from more previous best particles.

D.2(3) Variantes de PSO

Other PSO variants

- Binary/discrete particle swarms
- Constricted coefficients PSO
- PSO for noisy fitness functions
- PSO for dynamical problems
- PSO for multi-objective optimization problems
- Adaptive particle swarms
- PSO with diversity control
- Hybrids (e.g. with evolutionary algorithms)

D.2(3) Variantes de PSO

Dynamic Optimisation Problems

- **Originally developed for *static* optimization problems, the PSO algorithm has been adapted for the *dynamic* case by adding mechanisms to:**

  1) modify the social influence to maintain diversity in the optimization space and detect several optima;
  2) detect changes in the objective function by using the memory of each particle; and
  3) adapt the memory of its population if change occur in the optimization environment

D.2(3) Variantes de PSO

Dynamical Niching PSO (DNPSO)

- **This algorithm maintains diversity in the search space by:** [Nickabadi, 2008]
  1) using a local neighborhood topology, where sub-swarms are dynamically created around masters (particles that are their own local best in their neighborhood),
  2) defining a minimal distance within which two masters cannot co-exist,
  3) allowing free particles that do not belong to a sub-swarm, to move independently, and
  4) reinitializing those free particles that exhibit low velocities, meaning that they have converged on a non-optimal position.

D.2(3) Variantes de PSO

Multi-objective optimization

- **Originally developed for static *mono-objective* optimization, the PSO algorithm is adapted for *multi-objective* optimization problems with mechanisms to :**
  1) select and update of leaders
  2) promote diversity in the creation of new solutions using PSO position update and mutation operators
- **Algorithms for MOPSO are classified as: aggregating, lexicographic ordering, sub-population, Pareto-based, or combined approaches.**

D.2(3) Variantes de PSO

Multi-objective optimization

- Algorithms for multi-objective optimization aim to generate and select a set of **non-dominated solutions** (that belong to a Pareto front), instead of a single solution as in global optimization:
- **In multi-objective PSO problems:**

- Each particle may have a different set of leaders from which just one can be selected to update its position. The set of leaders is stored in an **external archive** of non-dominated solutions.
- The solutions contained in the archive are used as leaders to update particle positions, and are also reported as the final output of the algorithm.

D.2(3) Variantes de PSO

Multi-objective optimization

- **MOPSO algorithms:**

1) The swarm is first initialized. A set of leaders is also initialized with the non-dominated particles from the swarm, and stored in an external archive.

2) During each generation and for each particle, a leader is selected, and the particle position is updated.

3) The particle's fitness is then evaluated and its corresponding *pbest* value is updated. A new particle usually replaces its *pbest* particle when this particle is dominated or if both are non-dominated with respect to each other.

D.2(3) Variantes de PSO

Multi-objective optimization

- **MOPSO algorithms:**
  - **Pareto-based approaches** use leader selection techniques based on Pareto dominance [Coello Coell, 2008].
  - Leaders are defined as particles that are non-dominated with respect to the swarm.
  - Most authors adopt additional information (e.g., information provided by a density estimator) in order to avoid a random selection of a leader from the current set of non-dominated solutions.

CONTENU DU COURS

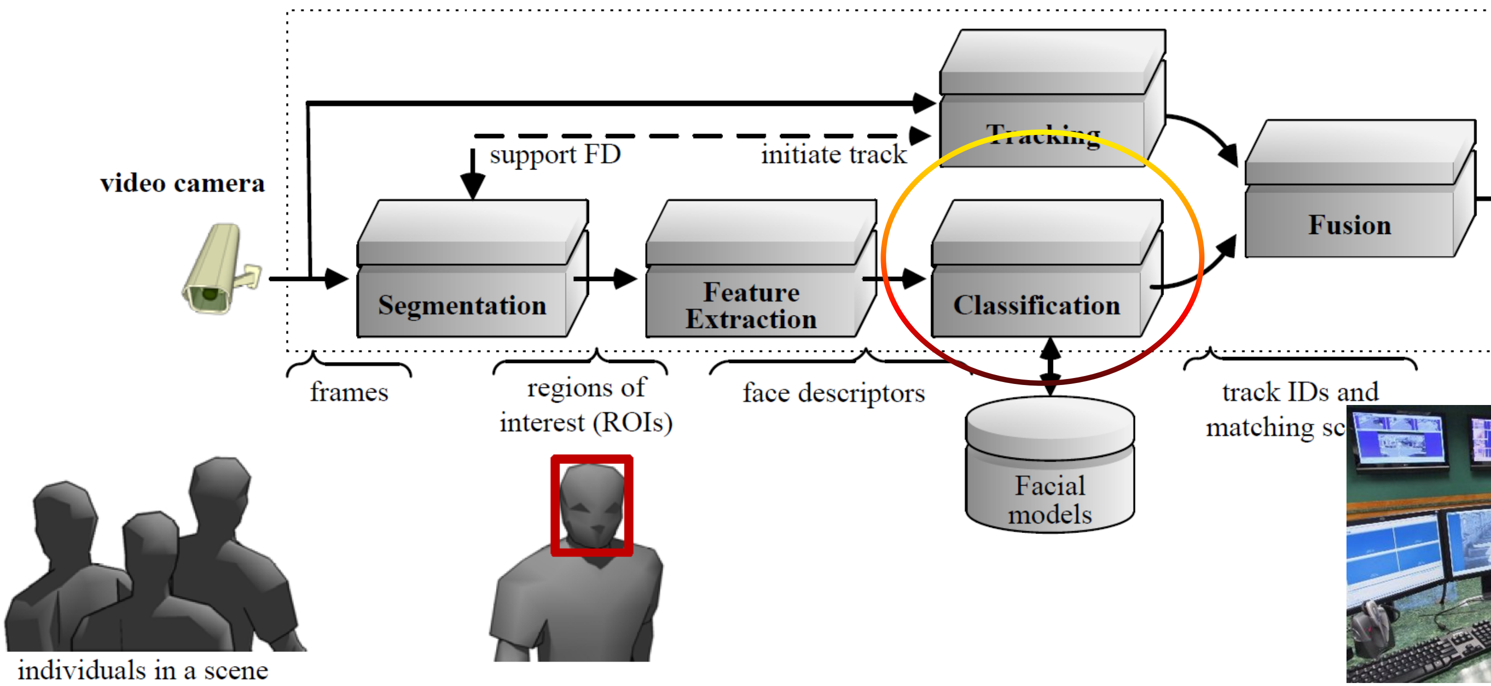
D.2 Optimisation par essaims particulaires

1) Intelligence d'essaims
2) Algorithme PSO canonique
3) Variantes de PSO
   – Environnements dynamiques
   – Optimisation multicritère
4) **Application – Optimisation évolutionnaire de RNA**

D.2(4) Optimisation du FAM

Reconnaissance de visages en vidéosurveillance

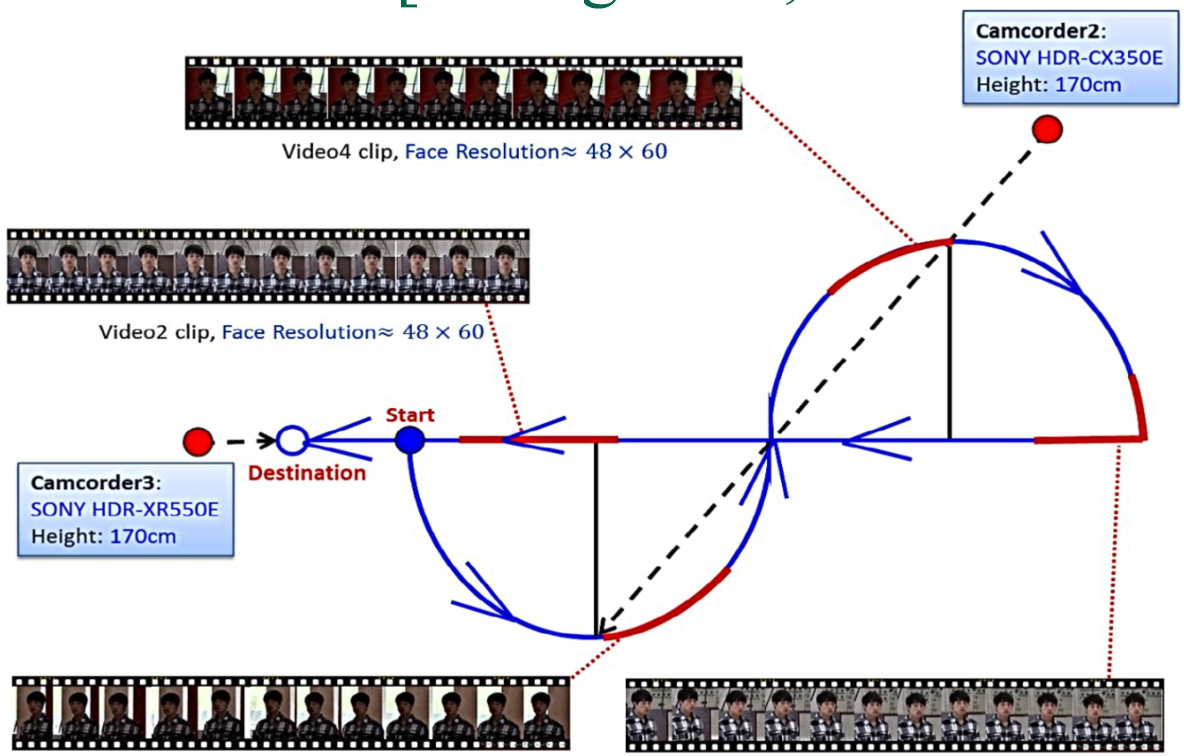
- **Système générique pour la reconnaissance spatio-temporelle de visages en vidéo surveillance**



D.2(4) Optimisation du FAM

Reconnaissance de visages en vidéosurveillance

- **Base COX-S2V:** les individus marchent à travers un circuit de caméras [Huang et al., ACCV 2012]



D.2(4) Optimisation du FAM

Reconnaissance de visages en vidéosurveillance

- **Base COX-S2V:** les individus marchent à travers un circuit de caméras [Huang et al., ACCV 2012]

statique

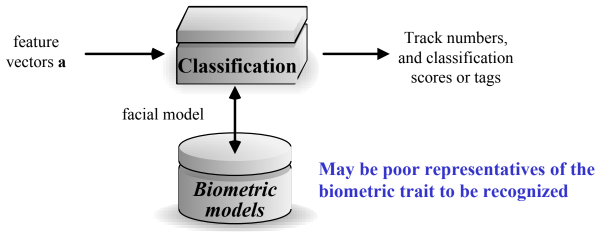
vidéo

D.2(4) Optimisation du FAM

Reconnaissance de visages en vidéosurveillance

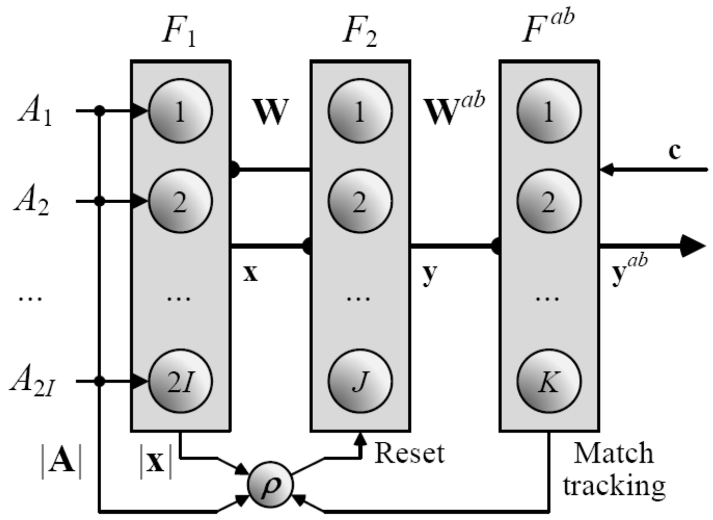
- **Défis – ressources de calcul:** les réseaux de vidéo surveillance comprennent beaucoup de caméras

- **Défis – environnements réelles sont complexes et changent dynamiquement:**
  - la compression et basse qualité et résolution des vidéos
  - interopérabilité des caméras
  - **conditions d'acquisition:** variations de pose, expression, occlusion, illumination, échelle, floue, etc.
  - **modèles de visages peu robustes:**
    conçus a priori (lors de l'abonnement)
    avec des ROI référence en nombre limité



Réseau de classification ARTMAP

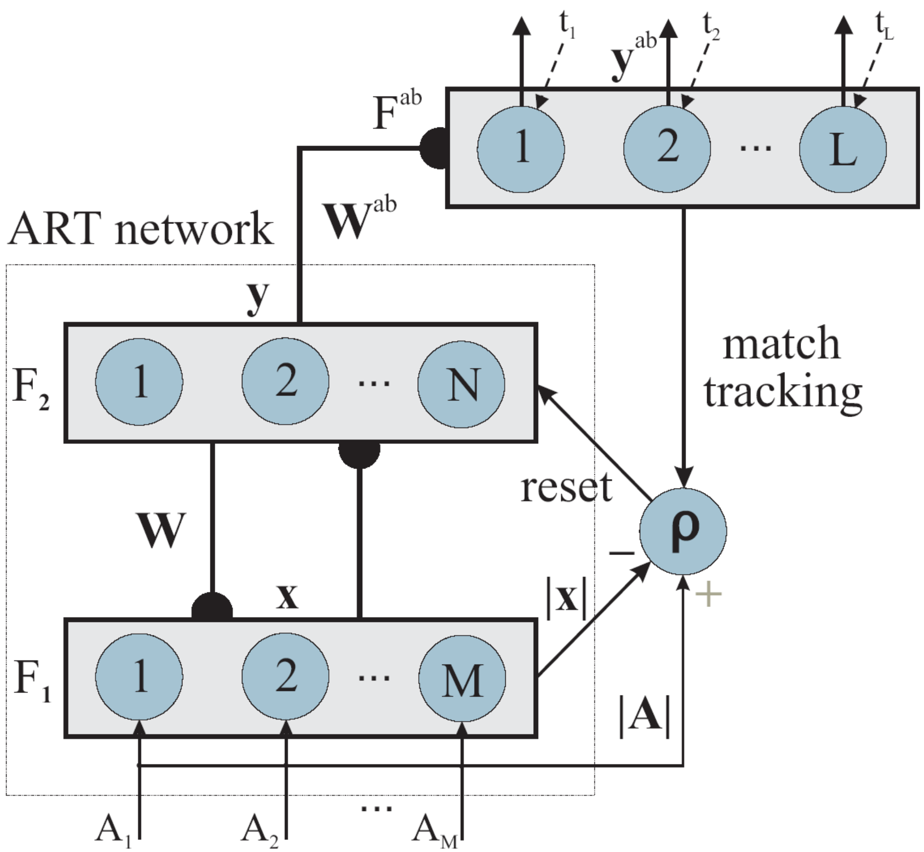
- **Versatility:** capables d'apprentissage rapide, en-ligne, supervisé, non-supervisé et incrémental
- **Constructive:** les poids et l'architecture (neurones *F2*) peuvent s'adapter en fonction de nouvelles données



D.2(4) Optimisation du FAM

Réseau de classification fuzzy ARTMAP

- **Structure simplifiée d'un réseau ARTMAP:**



D.2(4) Optimisation du FAM

Réseau de classification fuzzy ARTMAP

- **Algorithme – mode entraînement:**
  1. **Initialisation des poids:** fixer tous les poids $\mathbf{W}^{ab} = 0$
  2. **Encodage de la prochaine entrée: (a, t)**
  3. **Remise à zéro du seuil de vigilance** ρ
  4. **Choix d'une catégorie**
  5. **Applique le critère de vigilance**
  6. **Prédiction d'une classe:**
     - le code de réponse désirée **t** est présenté à F^{ab}
     - **fonction de prédiction:** le patron **y** active la couche F^{ab} via les poids $\mathbf{W}^{ab}$ $$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N} y_j w_{jk}^{ab}$$

D.2(4) Optimisation du FAM

Réseau de classification fuzzy ARTMAP

- **Algorithme – mode entraînement:**

6. Prédiction d'une classe: (suite)

- prédiction: $K = \max\left\{S_k^{ab}(\mathbf{y}) : k = 1, 2, \ldots, L\right\}$

  code binaire $\mathbf{y}^{ab}$ est actif pour le neurone K correspondant à la prédiction ($y_K^{ab} = 1$ et $y_k^{ab} = 0$ pour $k \neq K$)

- si la prédiction K correspond à la réponse désirée, on procède à l'apprentissage (étape 7), sinon on effectue un *'match tracking'*

D.2(4) Optimisation du FAM

Réseau de classification fuzzy ARTMAP

- **Algorithme – mode entraînement:**
  - ‘***match tracking***’:

$$\rho' = (|\mathbf{A} \wedge \mathbf{w}_J|/M) + \varepsilon$$

augmente ρ du fuzzy ART juste assez pour induire une nouvelle recherche pour soit:

- trouver un autre neurone commis de F2 qui prédit la classe désirée (étape 4)
- initier un neurone non-commis de F2 pour apprendre la classe désirée (étape 7)

D.2(4) Optimisation du FAM

Réseau de classification fuzzy ARTMAP

- **Algorithme – mode entraînement:**

7. Apprentissage:

- **mise à jour du prototype de *J*:** le vecteur prototype $\mathbf{w}_J$ du neurone J est adapté selon:

$$\mathbf{w}'_J = \beta(\mathbf{A} \wedge \mathbf{w}_J) + (1 - \beta)\mathbf{w}_J$$

- **création d'un nouveau lien associatif:** si J vient d'être commis, on fixe $\mathbf{w}_{JK}^{ab} = 1$, où $k = K$ est la réponse désiré

Retour à l'étape 2 pour prendre une autre entrée

D.2(4) Optimisation du FAM

Réseau de classification fuzzy ARTMAP

- **Algorithme – mode test:**

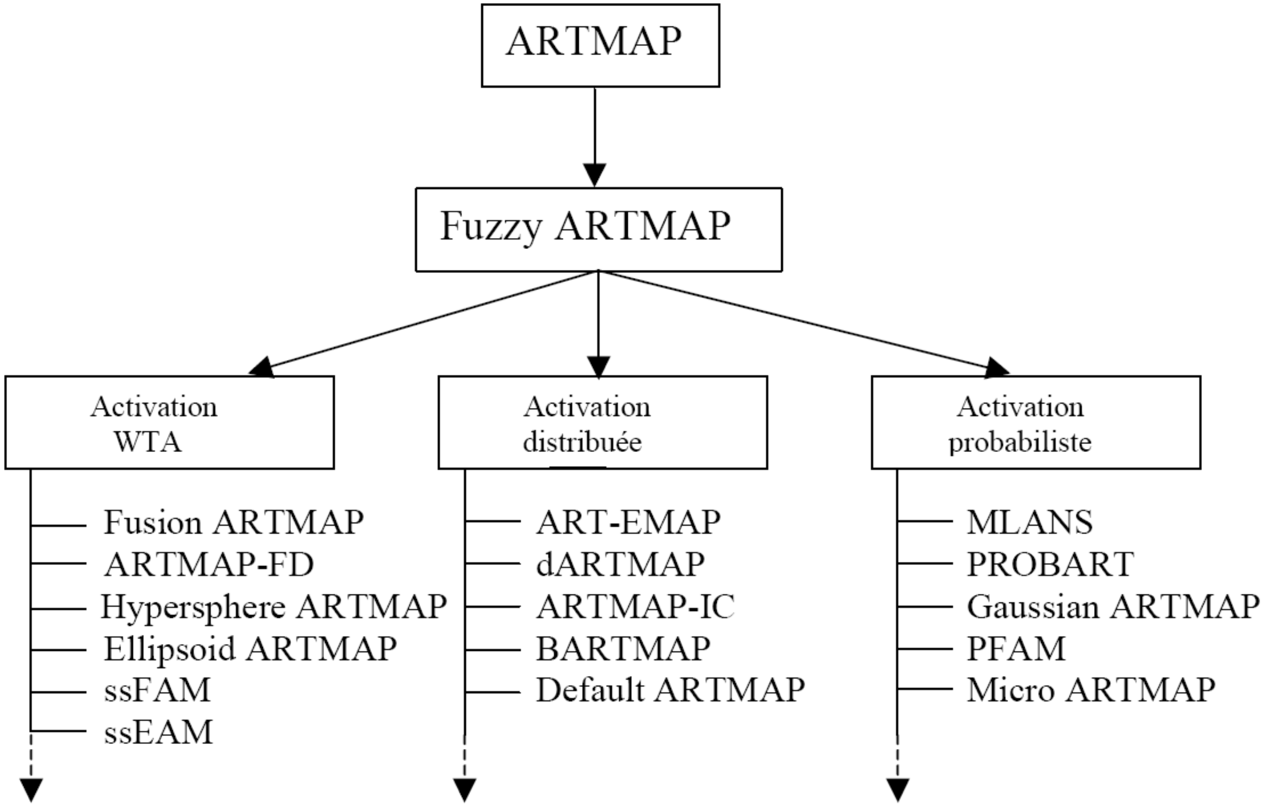
Afin de prédire la classe liée à chaque patron d'entrée:

1.
2. Encodage d'un patron d'entrée **a**
3.
4. Choix de catégorie
5.
6. Prédiction d'une K classe (sans tests)
7.

D.2(4) Optimisation du FAM

Réseau de classification fuzzy ARTMAP

- Une taxonomie des RNA de la famille ARTMAP



D.2(4) Optimisation du FAM

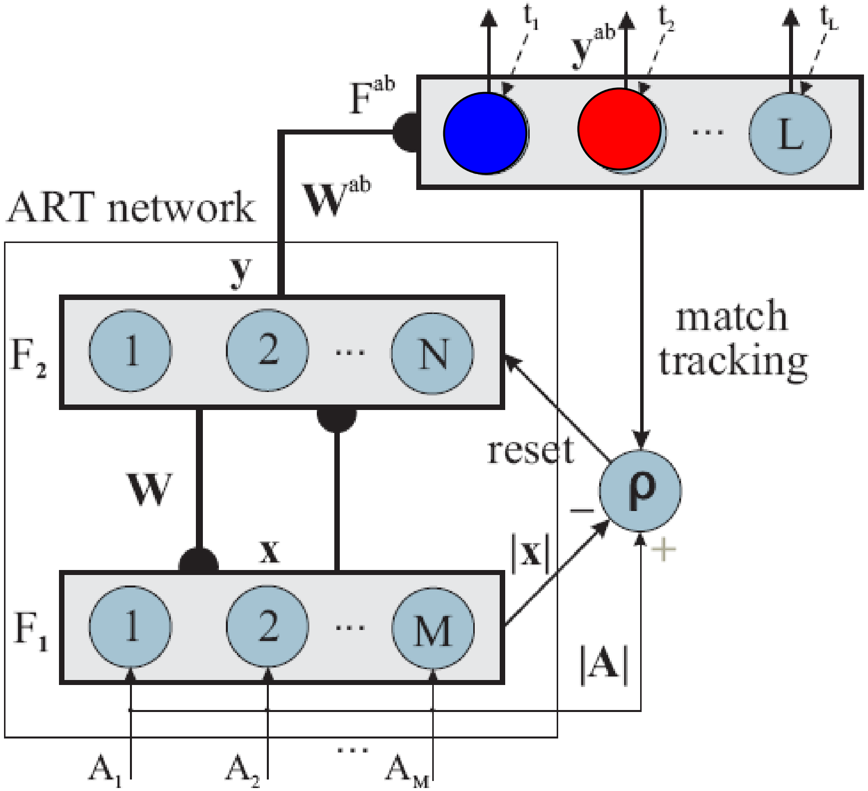
Taxonomy of the ARTMAP architecture (based on the internal matching process)

1. **Fuzzy category activation:**
   - a class is represented by one or more fuzzy set hyper-rectangles
   - category activation using Webber law choice function
   - EX: **fuzzy ARTMAP**, ART-EMAP, distributed ARTMAP, ARTMAP-IC…

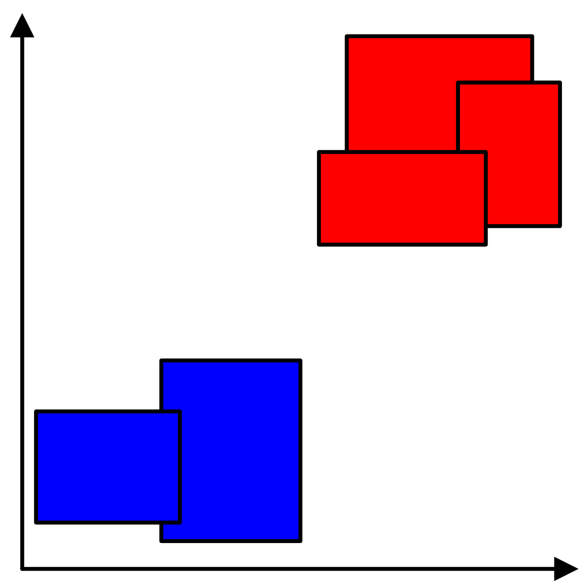
2. **Probabilistic category activation:**
   - a class is represented by one or more normal density functions
   - estimate the posterior probability of each class in order to apply the Bayes decision procedure
   - EX: PROBART, PFAM, MLANS, **Gaussian ARTMAP**, Ellipsoid ARTMAP, boosted ARTMAP

D.2(4) Optimisation du FAM

Fuzzy ARTMAP

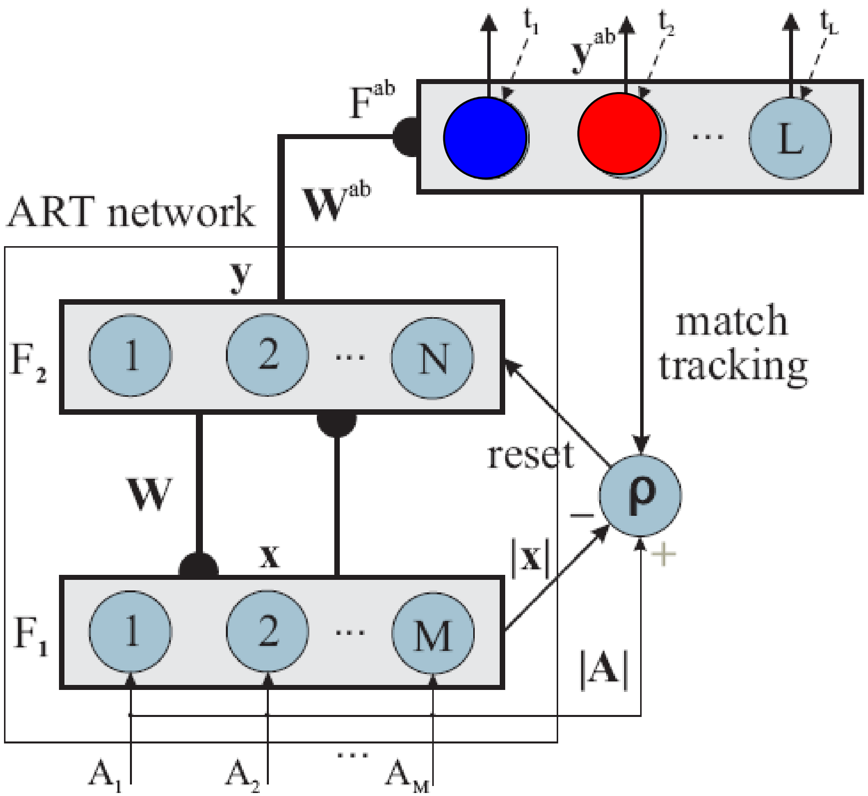


- category (F_2 node) ≡ a fuzzy set hyper-rectangle
- class (F^{ab} node) ≡ 1+ fuzzy set hyper-rectangles

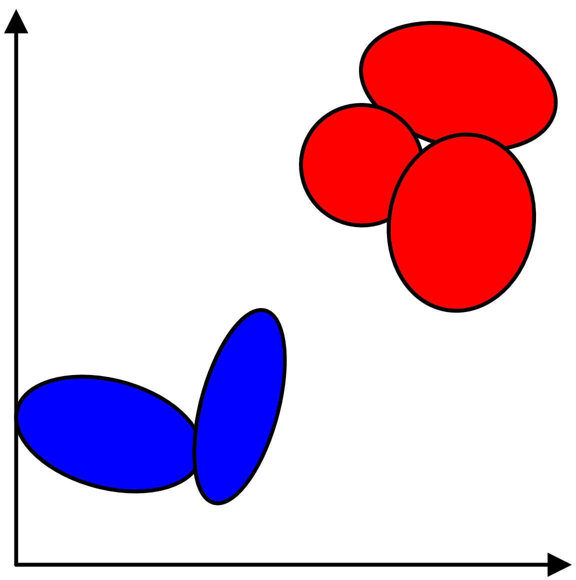


D.2(4) Optimisation du FAM

Gaussian ARTMAP



- category (F_2 node) ≡ a normal distribution
- class (F^{ab} node) ≡ 1+ normal distribution



D.2(4) Optimisation du FAM

TABLE I

EQUATIONS USED BY THE ARTMAP NETWORKS. WITH FUZZY ARTMAP, $|\cdot|$ IS THE L^1 NORM OPERATOR ($|\mathbf{w}_j| \equiv \sum_{i=1}^{M} |w_{ji}|$), $\wedge$ IS THE FUZZY AND OPERATOR ($(\mathbf{A} \wedge \mathbf{w}_j)_i \equiv \min(A_i, w_{ji})$), AND α, β, ε AND $\bar{\rho}$ ARE THE CHOICE, LEARNING RATE, MATCH TRACKING AND BASELINE VIGILANCE PARAMETERS, RESPECTIVELY. WITH GAUSSIAN ARTMAP, γ IS THE INITIAL STANDARD DEVIATION ASSIGNED TO NEWLY-COMMITTED F_2 NODES.

Algorithmic step	**ARTMAP neural network**	
	fuzzy ARTMAP	Gaussian ARTMAP
1. Initialization:	$\alpha > 0,\ \beta \in [0,1],\ 0 < \varepsilon \ll 1,\ \bar{\rho} \in [0,1]$	$\gamma > 0,\ 0 < \varepsilon \ll 1,\ \bar{\rho} \in [0,1]$
2. Input pattern coding:	$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)\ (M = 2m)$	$\mathbf{A} = \mathbf{a}\ (M = m)$
3. Prototype selection:		
– choice function	$T_j(\mathbf{A}) = \lvert\mathbf{A} \wedge \mathbf{w}_j\rvert / (\alpha + \lvert\mathbf{w}_j\rvert)$	$g_j(\mathbf{A}) = \begin{cases} \frac{n_j}{\prod_{i=1}^{M} \sigma_{ji}} G_j(\mathbf{A}) & \text{if } G_j(\mathbf{A}) > \rho \\ 0 & \text{otherwise} \end{cases}$
– vigilance test	$\lvert\mathbf{A} \wedge \mathbf{w}_J\rvert \geq \rho m$	$G_j(\mathbf{A}) = \exp\left\{ -\frac{1}{2} \sum_{i=1}^{M} \frac{(A_i - \mu_{ji})^2}{\sigma_{ji}^2} \right\} > \rho$
– F_2 activation	$y_j = 1$ only if $j = J$ (winning node)	$y_j = g_j / (0.01 + \sum_{l=1}^{N_c} g_l)$
4. Class prediction:		
– prediction function	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N} y_j w_{jk}^{ab}$	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N_c} y_j w_{jk}^{ab}$
– match tracking	$\rho' = (\lvert\mathbf{A} \wedge \mathbf{w}_J\rvert / m) + \varepsilon$	$\rho' = \exp\left\{ -\frac{1}{2} \sum_{j \in E_K} y_j^* \sum_{i=1}^{M} \frac{(A_i - \mu_{ji})^2}{\sigma_{ji}^2} \right\} + \varepsilon$
5. Learning:		
– prototype update	$\mathbf{w}'_J = \beta(\mathbf{A} \wedge \mathbf{w}_J) + (1 - \beta)\mathbf{w}_J$	$n'_j = n_j + y_j^*$ $\mu'_{ji} = (1 - \frac{y_j^*}{n_j})\mu_{ji} + \frac{y_j^*}{n_j} A_i$ $\sigma'_{ji} = \sqrt{(1 - \frac{y_j^*}{n_j})\sigma_{ji}^2 + \frac{y_j^*}{n_j}(A_i - \mu_{ji})^2}$

D.2(4) Optimisation du FAM

Réseau ARTMAP-FD

- **Entraînement sur patrons de classes connues:**

patrons	étiquette de classe
A *a* a	class #1
b b ***B***	class #2
C c **C**	class #3

- **Test sur patrons de classes connues *et* inconnues:**

patrons	prédiction du classificateur
A	class #1
C	class #3
D	***classe inconnue!***

D.2(4) Optimisation du FAM

Réseau ARTMAP-FD

- **ARTMAP-FD:** une extension de fuzzy ARTMAP qui permet de détecter des patrons qui appartiennent à de classes inconnues
  - pour chaque entrée **a** en mode test, on calcul la mesure de familiarité:

$$\phi(\mathbf{A}) = \frac{T_J(\mathbf{A})}{T_J^{\max}(\mathbf{A})} = \frac{|\mathbf{A} \wedge \mathbf{w}_J|}{|\mathbf{w}_J|}$$

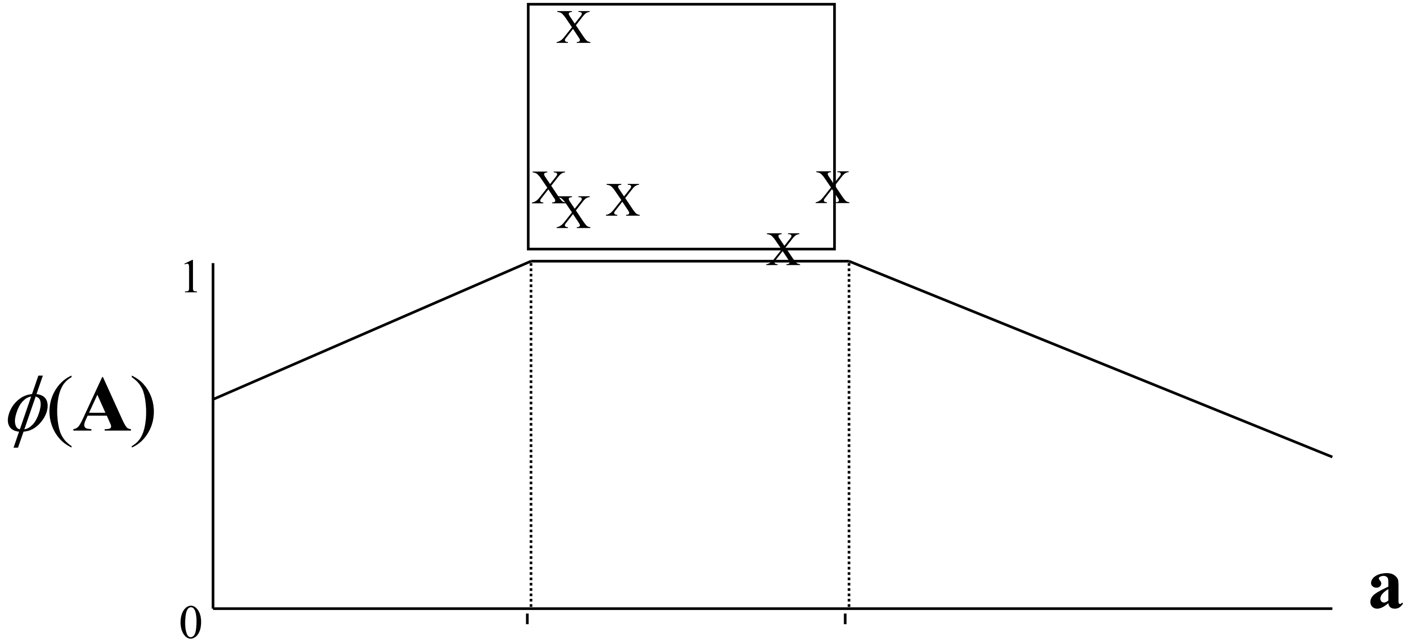
Si $\phi(\mathbf{A}) > \gamma \longrightarrow$ patron déclaré connu (prédit une classe K)

Si $\phi(\mathbf{A}) \leq \gamma \longrightarrow$ patron déclaré inconnu (aucune prédiction)

D.2(4) Optimisation du FAM

Réseau ARTMAP-FD

- **Mesure simple du degré d'appartenance d'un patron à un hyper rectangle (catégorie):**
  - φ(**A**) = 1 à l'intérieur, et φ(**A**) < 1 à l'extérieur



D.2(4) Optimisation du FAM

Common training strategies for FAM

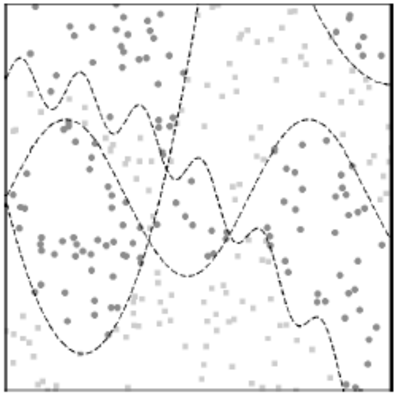
1. **one epoch (`1EP`):**
   - learning is completed after one epoch
2. **convergence based on training set classifications (`CONVp`):**
   - learning ends once no training patterns are misclassified
3. **convergence based on weight values (`CONVw`):**
   - learning ends once weights remain constant for 2 successive epochs
4. **hold-out validation between epochs (`HV`):**
   - learning ends once the E_{gen} is minimized on an independent validation subset

D.2(4) Optimisation du FAM

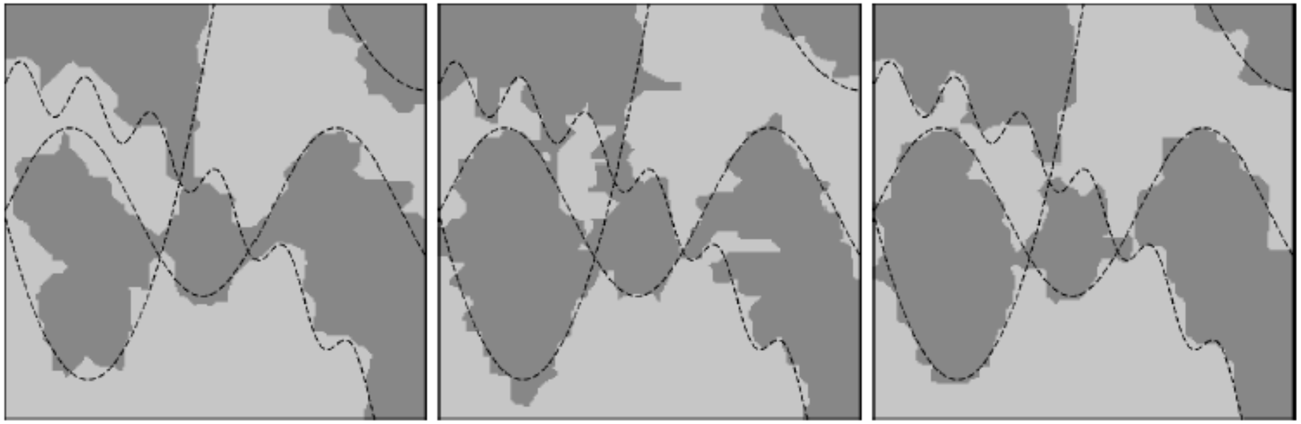
Réseau de classification fuzzy ARTMAP

- **Training dynamics is governed by 4 inter-related hyperparameters:** $\mathbf{h} = (\beta,\ \alpha,\ \bar{\rho},\ \varepsilon)$
  - **Example:** decision boundaries on P2 data for FAM trained with different hyperparameters values **h**:

$\mathbf{h}_1 = (0.7;\ 0.7;\ 0.8;\ 0.85)$ $\mathbf{h}_2 = (0.13;\ 0.41;\ 0.08;\ 0.86)$ $\mathbf{h}_3 = (0.67;\ 0.73;\ 0.68;\ 0.89)$



(a) Original data



(b) FAM decision boundaries

D.2(4) Optimisation du FAM

PSO learning strategy

[Granger *et al.*, JPRR, 2007]

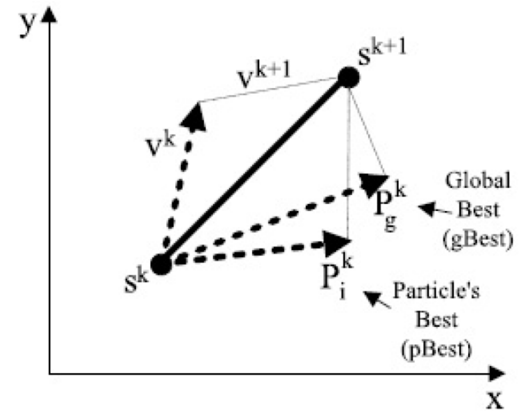
- **Mono-objective:** maximize the FAM classification rate in the hyperparameter space, **h**
  - **PSO:** population-based evolutionary optimization technique
  - **swarm** ≡ a population or pool of *N* particles, each one corresponding to a FAM network evolving in the **h** space
  - **PSO training strategy:** co-jointly determines all parameters of a FAM network (weights + architecture + **h)** such that E_{gen} is minimized

D.2(4) Optimisation du FAM

PSO learning strategy

- **Inspired by the synchronous parallel version of PSO, with exchange of information using:**
  1. ***lbest***: $\mathbf{p}_i^k$ – the best previously-visited position of particle *i*
  2. ***gbest***: $\mathbf{p}_g^k$ – the best particle position for the swarm
- **PSO update of $\mathbf{h}_i^k$ in 2-D space for iteration *k*+1:**
  – particles move though the search space by following the particle with the best fitness value

$$\mathbf{h}_i^k = (\beta_i^k, \alpha_i^k, \bar{\rho}_i^k, \varepsilon_i^k)$$



D.2(4) Optimisation du FAM

PSO learning strategy

A. Initialization: N, $k_{\max}$, r_1, r_1, c_1, c_2, w^k, $\mathbf{h}_i^{0}$, etc.

B. Iterations:

while $k \leq k_{\max}$ or $E_{\text{gen}}(\mathbf{p}_g^{k}) - E_{\text{gen}}(\mathbf{p}_g^{k-1}) < \varphi$ do

- for $i = 1, 2, \ldots, N$ particles
  - train FAM network using parameters of $\mathbf{h}_i^k$
  - compute fitness value of network $E_{\text{gen}}(\mathbf{h}_i^k)$
  - if $E_{\text{gen}}(\mathbf{h}_i^k) \leq E_{\text{gen}}(\mathbf{p}_i^k)$, update ***lbest*** ($\mathbf{p}_i^k = \mathbf{h}_i^k$)
- select the ***gbest*** particle, $g = \text{argmin}\{E_{\text{gen}}(\mathbf{h}_i^k)\}$
- for $i = 1, 2, \ldots, N$ particles
  - update particle velocity $\mathbf{v}_i^{k+1}$ and position $\mathbf{h}_i^{k+1}$
- update particle inertia $\boldsymbol{w}^k$, and increment $k = k + 1$

D.2(4) Optimisation du FAM

PSO learning strategy

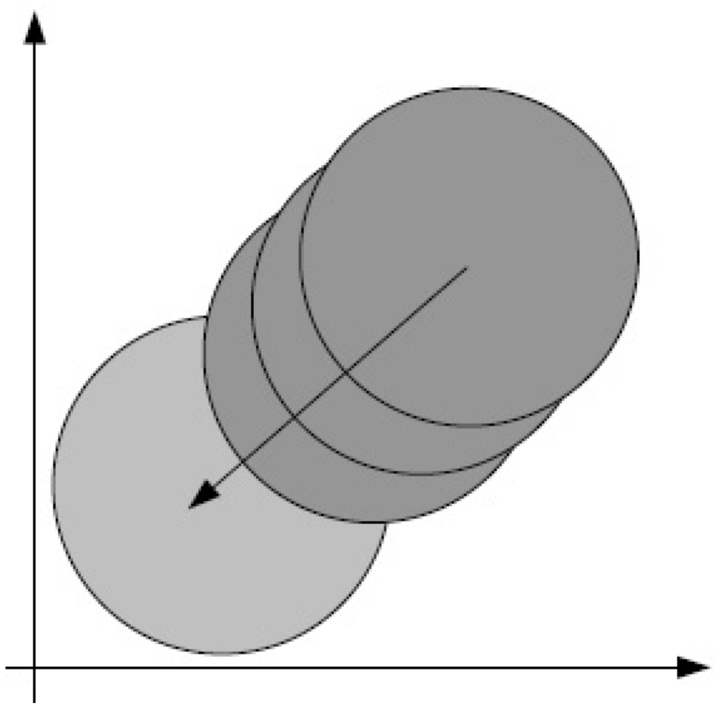
- **Experimental Methodogy:**
  - **4 independent replications:** retain network with best $\mathbf{p}_g^k$
  - **each replication is performed with $N = 15$ particles**
    - particle vectors are initialized randomly according to a uniform distribution
    - except $\mathbf{h}_i^0$ is set to minimize resources
  - **a trial ends if:**
    - $k_{\max} = 100$ iterations
    - $E_{\text{gen}}(\mathbf{p}_g^k)$ s constant for 10 consecutive iterations
  - $c_1 = c_2 = 2$
  - w^k **decreased linearly from 0.9 to 0.4 over** $k_{\max}$
  - r_1 **and** r_1 **= random numbers from a uniform distribution**

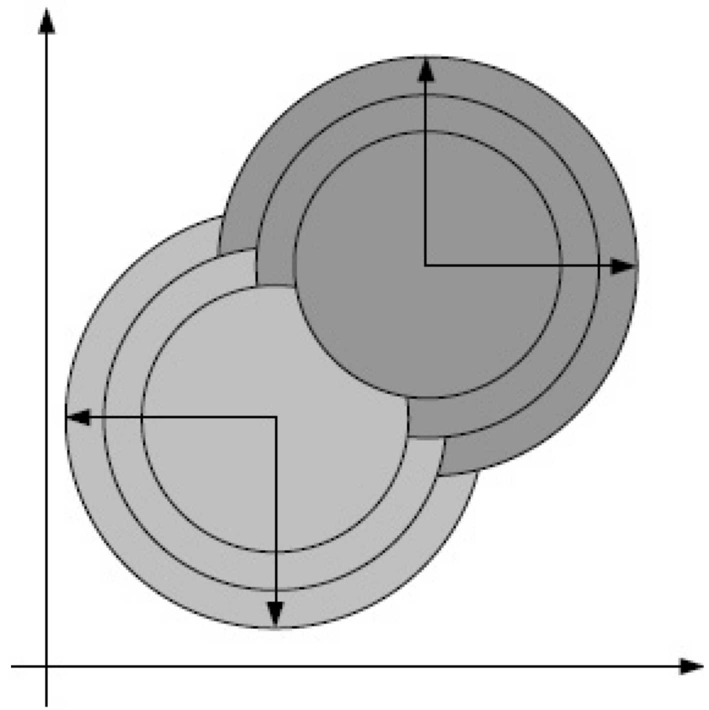
D.2(4) Optimisation du FAM

PSO learning strategy

- **Synthetic data sets $D_\mu(\xi_{tot})$ and $D_\sigma(\xi_{tot})$** – linear decision bounds where class distributions overlap:

$D_\mu(\xi_{tot})$ $D_\sigma(\xi_{tot})$



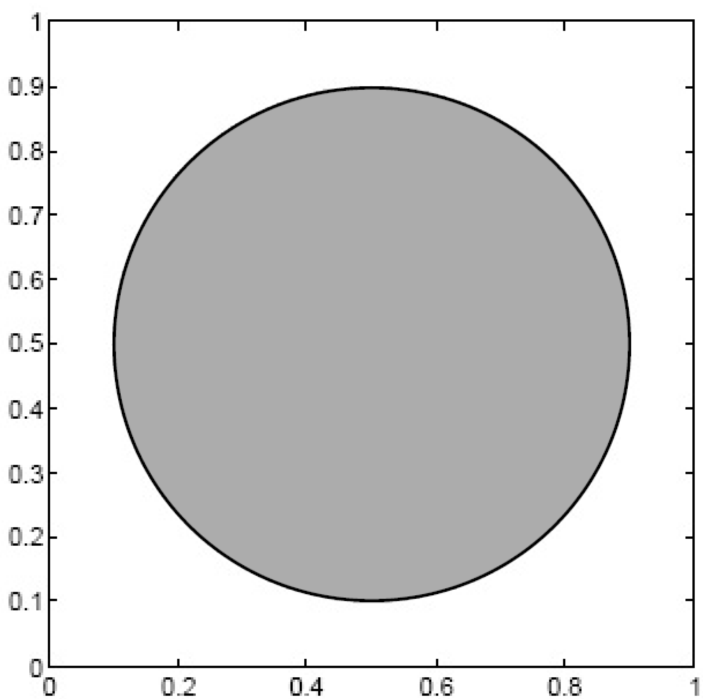


D.2(4) Optimisation du FAM

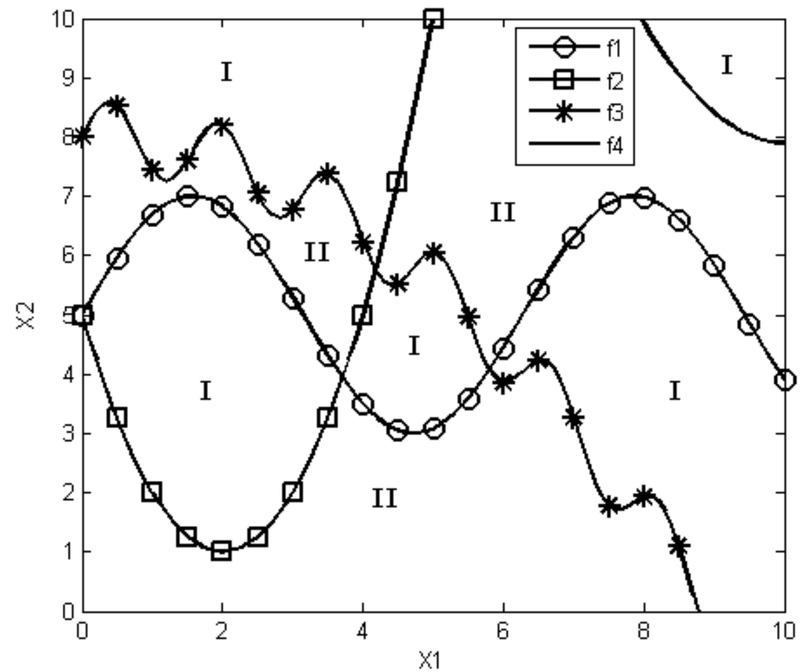
PSO learning strategy

- **Synthetic data sets D_{CIS} and D_{P2}** – non-linear decision bounds where class distributions do not overlap:

D_{CIS}



D_{P2}

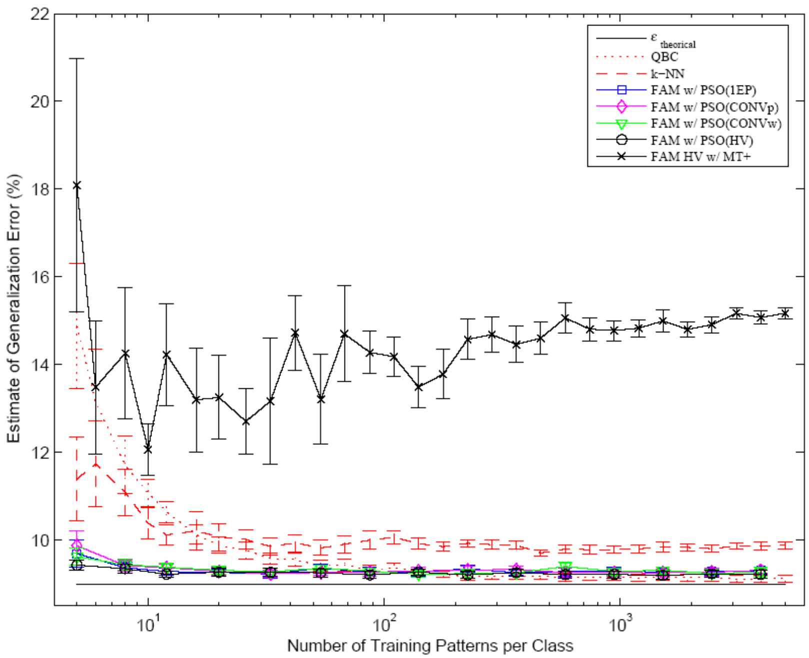


D.2(4) Optimisation du FAM

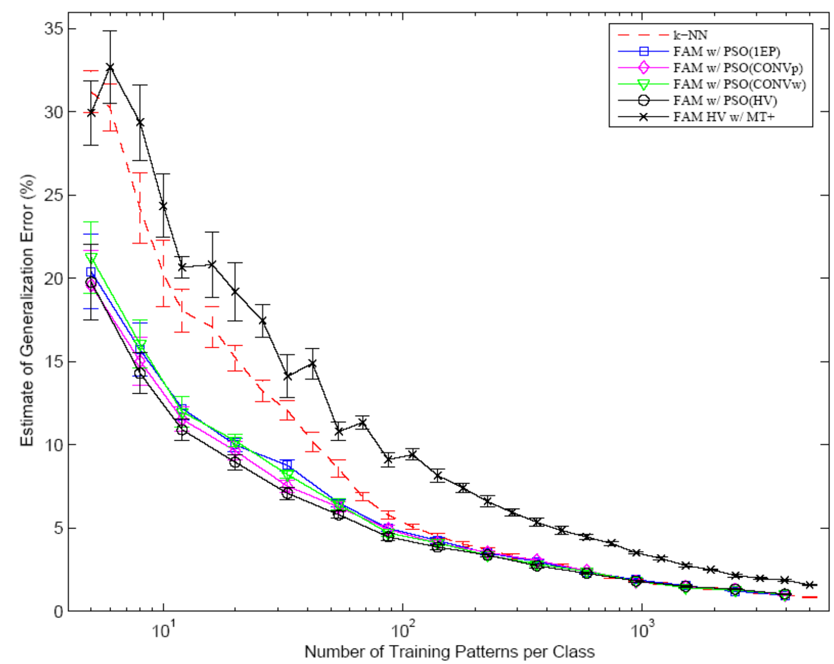
PSO learning strategy

- Average E_{gen} *vs.* training set size

$D_{\mu}(9\%)$



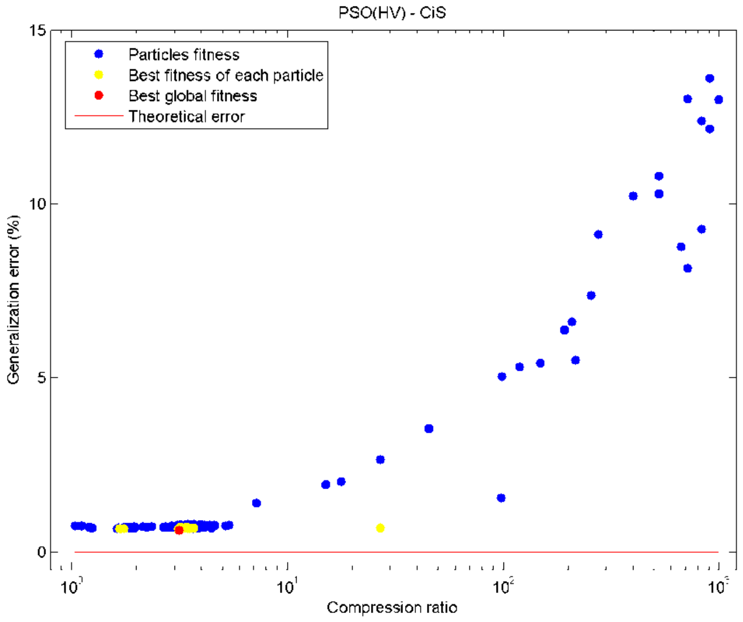
D_{CIS}



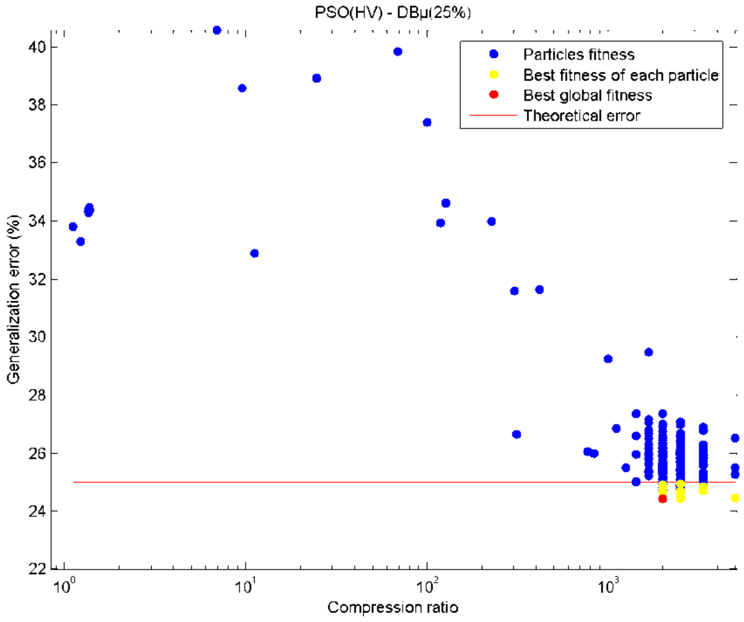
PSO learning strategy

- Average E_{gen} *vs.* network compression

D_{CIS}



$D_{\mu}(9\%)$



D.2(4) Optimisation du FAM

PSO learning strategy

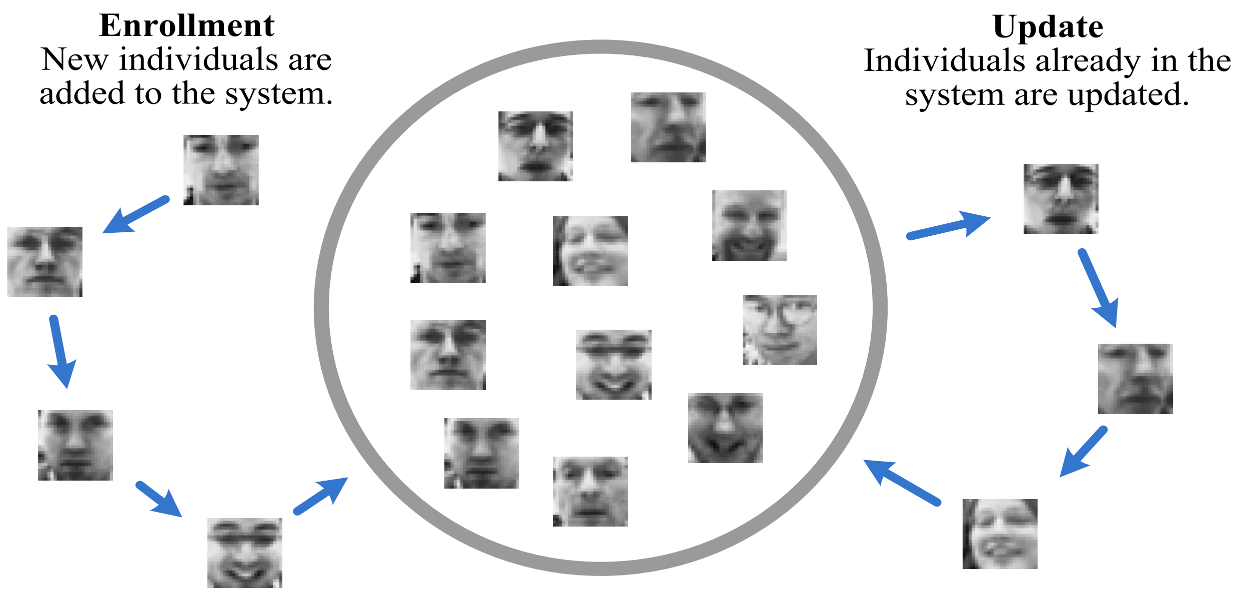
- **Average E_{gen} at 5000 patterns per class**

Classifier	Average generalization error (%)					
	$D_{\mu}(1\%)$	$D_{\mu}(9\%)$	$D_{\mu}(25\%)$	$D_{\sigma}(9\%)$	D_{CIS}	D_{P2}
theoretical ξ_{tot}	1,00	9,00	25,00	9,00	0,00	0,00
Quadratic Bayes	1,00 (0.04)	9,12 (0.08)	25,11 (0.10)	9,04 (0.07)	N/A	N/A
***k*-NN**	1,08 (0.03)	9,88 (0.08)	27,23 (0.12)	9,66 (0.07)	0,86 (0.03)	1,65 (0.05)
1-NN	1,54 (0.03)	13,35 (0.10)	33,49 (0.17)	13,04 (0.14)	0,84 (0.02)	1,61 (0.04)
FAM 1EP (MT+)	2,51 (0.14)	18,78 (0.38)	38,81 (0.36)	17,98 (0.29)	3,98 (0.21)	7,33 (0.34)
FAM CONVp (MT+)	1,90 (0.07)	15,44 (0.15)	36,14 (0.20)	14.90 (0.06)	1,47 (0.05)	3,61 (0.08)
FAM CONVw (MT+)	1,97 (0.09)	15,30 (0.16)	35,94 (0.15)	14,79 (0.10)	1,64 (0.05)	3,66 (0.08)
FAM HV (MT+)	1,88 (0.05)	15,17 (0.13)	36,10 (0.20)	14,99 (0.12)	1,58 (0.05)	3,68 (0.08)
FAM HV (MT-)	2,17 (0.08)	20,80 (0.56)	39,83 (0.31)	20,13 (0.43)	1,69 (0.07)	4,26 (0.16)
FAM PSO(1EP)	1,04 (0.03)	9,35 (0.08)	25,50 (0.09)	9,18 (0.06)	1,35 (0.12)	2,05 (0.10)

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

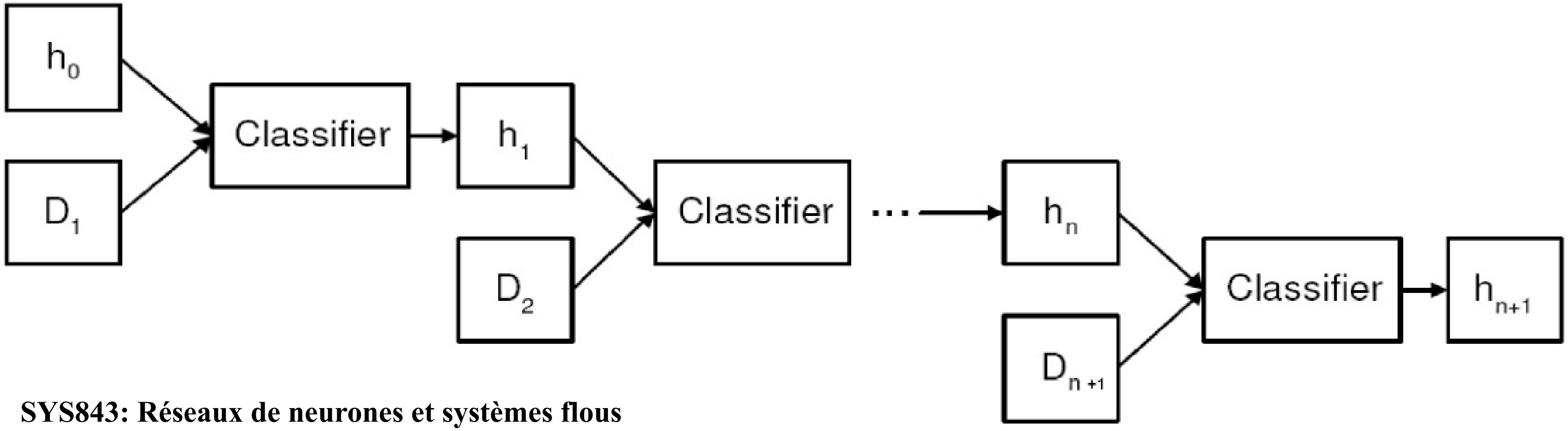
- **Abonnement et adaptation** de modèles faciales en fonction de nouvelles données référence



D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Supervised incremental learning**
  - new training data D_i is acquired from the environment at different instants in time t_i for $i = 1, 2, ..., n$
  - D_i : block of labeled training data available to the classifier at discrete time t_i
  - h_i : hypothesis of classifier based on h_{i-1} and training with D_i



D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

Survey – Incremental Learning Techniques

1. **Classifiers designed for incremental learning:** such as ARTMAP and Growing Self-Organizing families of neural networks

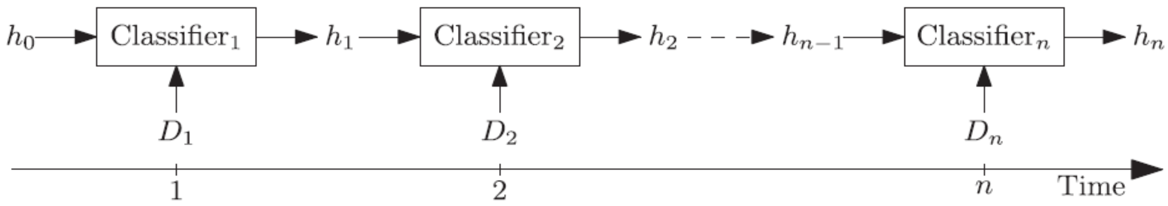
2. **Adaptations of popular classifiers:** such as the SVM, and the MLP and RBF neural networks

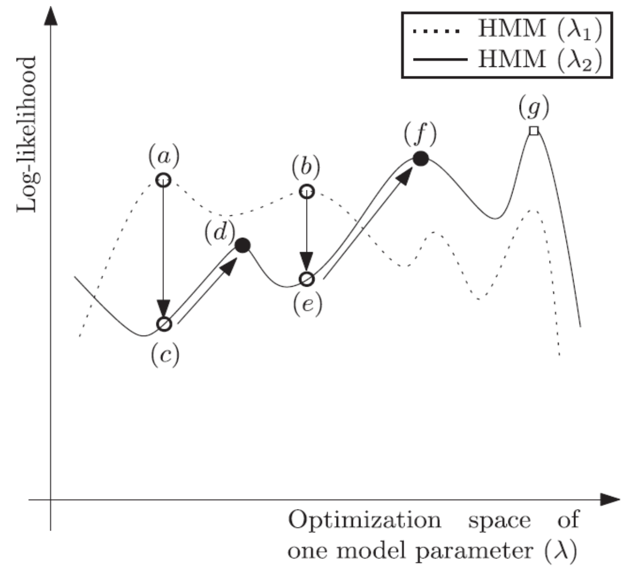
3. **Ensemble of Classifiers:** such as Learn++ with MLPs

D.2(4) Optimisation du FAM

Challenge of Adaptation

- **Knowledge corruption** – common during incremental learning of new reference data [Khreich et al., Information Sciences 2012]

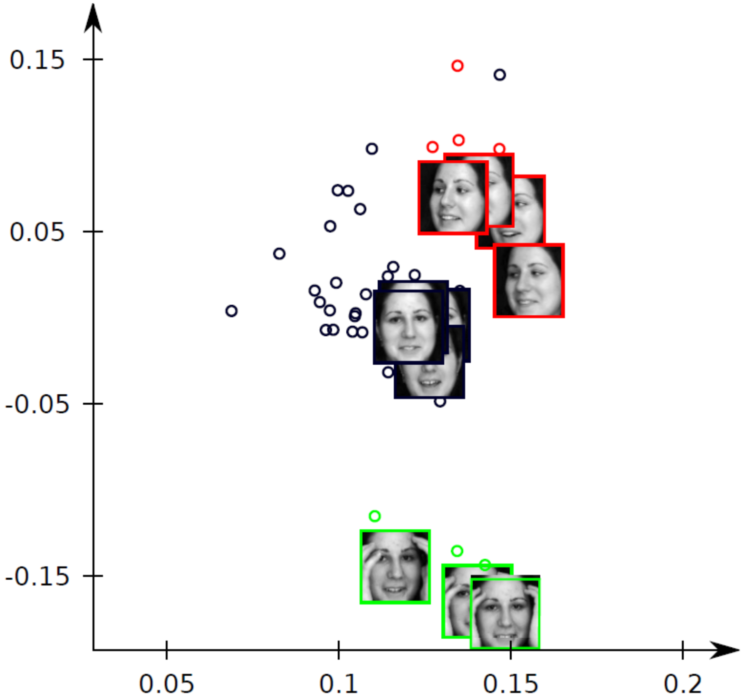




D.2(4) Optimisation du FAM

Challenge of Adaptation

- **La variations des conditions d'acquisition (e.g., illumination et pose) permet de définir différent *concepts:***

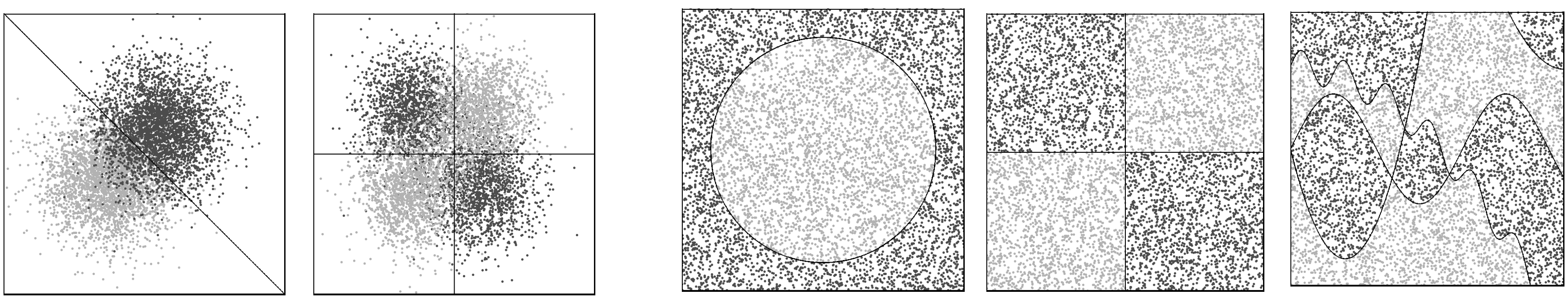


D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Data sets with complex decision boundaries and overlapping class distributions:**

D_{2N}(13%) D_{XOR}(13%) D_{CIS} $D_{XOR\text{-}U}$ D_{P2}



D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Protocole for each trial:**
  - divide each data set into LEARN and TEST subsets, each one with 5,000 patterns/class
  - subdivide LEARN into b blocks of data D_i ($i = 1, 2, \ldots, b$), each one with an equal number of patterns per class
    - large blocks: b =10 blocks with $|D_i|$ = 1000 patterns
    - small blocks: b =100 blocks with $|D_i|$ = 100 patterns
  - in each block, 2/3 of patterns from each class are used for training, and the rest for validation

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Batch learning process:**

t_1 : $\text{ARTMAP}_0(D_1)$ $\rightarrow \text{ARTMAP}_1$

t_2 : $\text{ARTMAP}_0(D_1 \cup D_2)$ $\rightarrow \text{ARTMAP}_2$

…

t_b : $\text{ARTMAP}_0(D_1 \cup D_2 \cup \ldots \cup D_b)$ $\rightarrow \text{ARTMAP}_b$

- **Incremental learning process:**

t_1 : $\text{ARTMAP}_0(D_1) \rightarrow \text{ARTMAP}_1$

t_2 : $\text{ARTMAP}_1(D_2) \rightarrow \text{ARTMAP}_2$

…

t_b : $\text{ARTMAP}_{b\text{-}1}(D_b) \rightarrow \text{ARTMAP}_b$

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

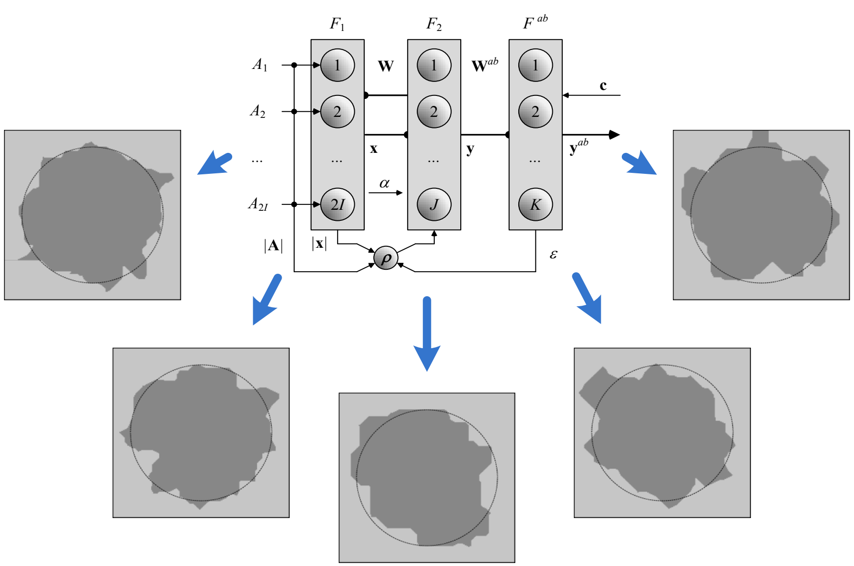
- **Average error rate of fuzzy and Gaussian ARTMAP networks with all data sets**

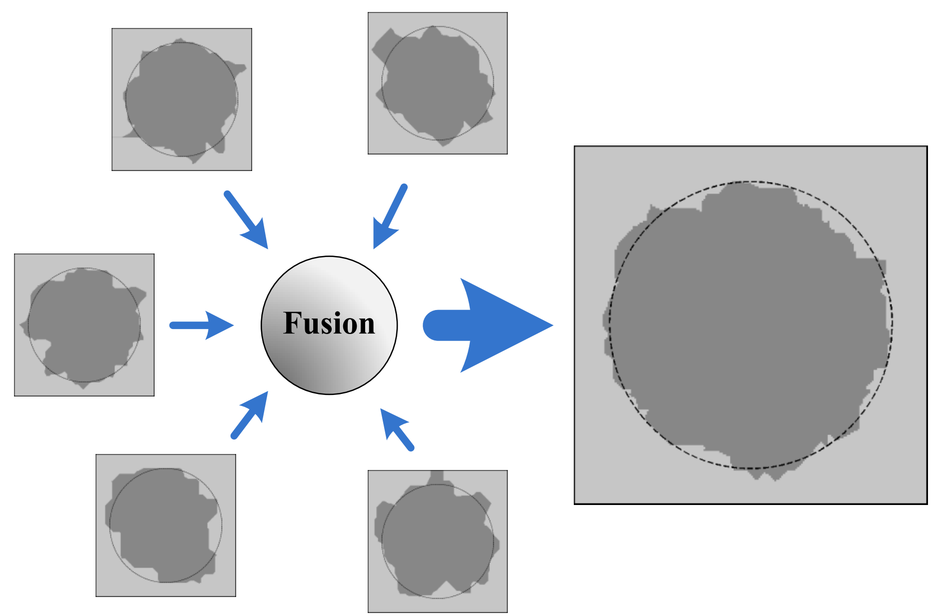
Data set	Average generalisation error (%)			
	fuzzy ARTMAP		Gaussian ARTMAP	
	batch	incremental $\lvert D_i\rvert = 1000 \rightarrow \lvert D_i\rvert = 100$	batch	incremental $\lvert D_i\rvert = 1000 \rightarrow \lvert D_i\rvert = 100$
$D_{2N}(1\%)$	2.1 (0.1)	3.2 (0.3) → 2.8 (0.2)	1.4 (0.1)	1.4 (0.1) → 1.4 (0.1)
$D_{2N}(13\%)$	21.6 (0.3)	23.8 (0.5) → 24.9 (0.6)	15.6 (0.5)	15.6 (0.6) → 15.4 (0.5)
$D_{2N}(25\%)$	35.8 (0.2)	37.8 (0.2) → 38.4 (0.2)	26.8 (0.2)	27.4 (0.2) → 27.3 (0.4)
$D_{3N}(1\%)$	2.4 (0.3)	3.0 (0.3) → 3.2 (0.4)	1.8 (0.6)	1.7 (0.3) → 1.5 (0.2)
$D_{XOR}(1\%)$	1.2 (0.1)	1.7 (0.2) → 1.7 (0.2)	2.6 (0.6)	3.7 (0.8) → 1.9 (0.6)
$D_{XOR}(13\%)$	25.3 (0.4)	27.6 (0.5) → 29.3 (0.5)	19.5 (0.7)	18.7 (0.6) → 18.6 (0.5)
$D_{XOR}(25\%)$	43.0 (0.3)	44.0 (0.3) → 44.0 (0.3)	35.7 (0.2)	37.4 (0.5) → 37.0 (0.3)
D_{CIS}	2.23 (0.05)	4.3 (0.2) → 4.7 (0.3)	1.9 (0.3)	1.9 (0.2) → 1.9 (0.1)
$D_{XOR\text{-}u}$	0.25 (0.07)	0.6 (0.1) → 0.9 (0.6)	1.4 (0.4)	1.9 (0.5) → 2.4 (0.6)
D_{P2}	5.11 (0.09)	7.9 (0.2) → 9.4 (0.4)	3.5 (0.3)	4.4 (0.3) → 5.9 (0.5)

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Approche pour adapter h = (α, β, ε, ρ):** stratégie d'apprentissage incrémentale basé sur DPSO, afin d'évoluer un ensemble hétérogène de classifieurs

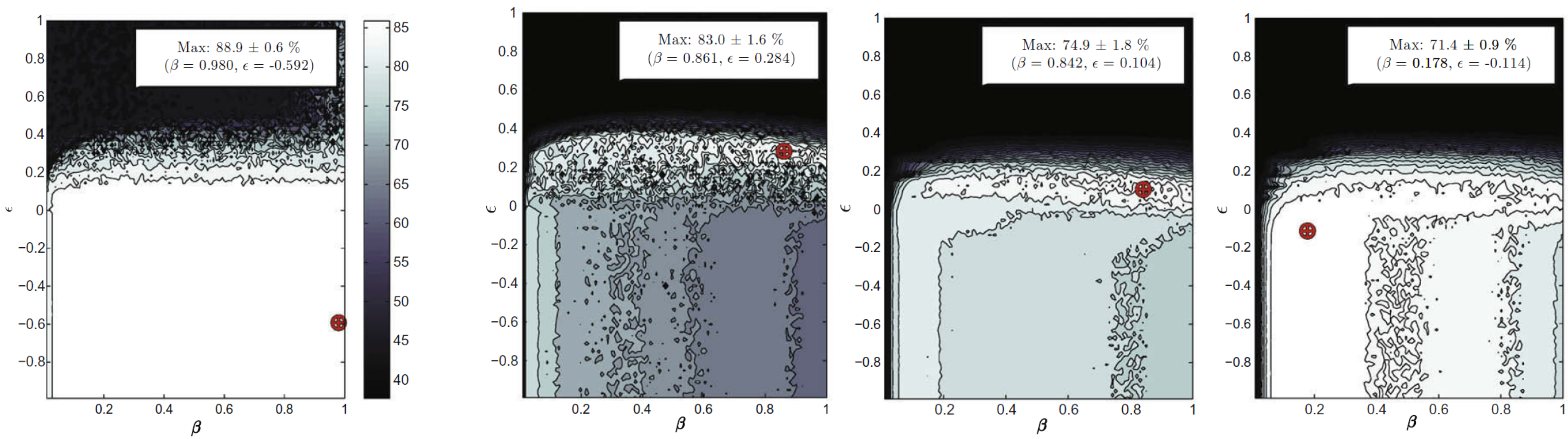




D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **L'appentissage incrémental de nouvelles données correspond à un problème d'*optimization dynamique* tel que:** $\text{maximize}\left\{ f(\mathbf{h}, t) \mid \mathbf{h} \in \mathbb{R}^4, t \in \mathbb{N}_1 \right\},$



(a) D_1 (b) D_2 (c) D_3 (d) D_4

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Un système adaptif pour le contrôle d'accès (*1:N*):**
- Comprend une LTM, un essaim de classifieurs incrémentaux et un module d'optimisation dynamique [Connolly *et al.*, PR 2012]

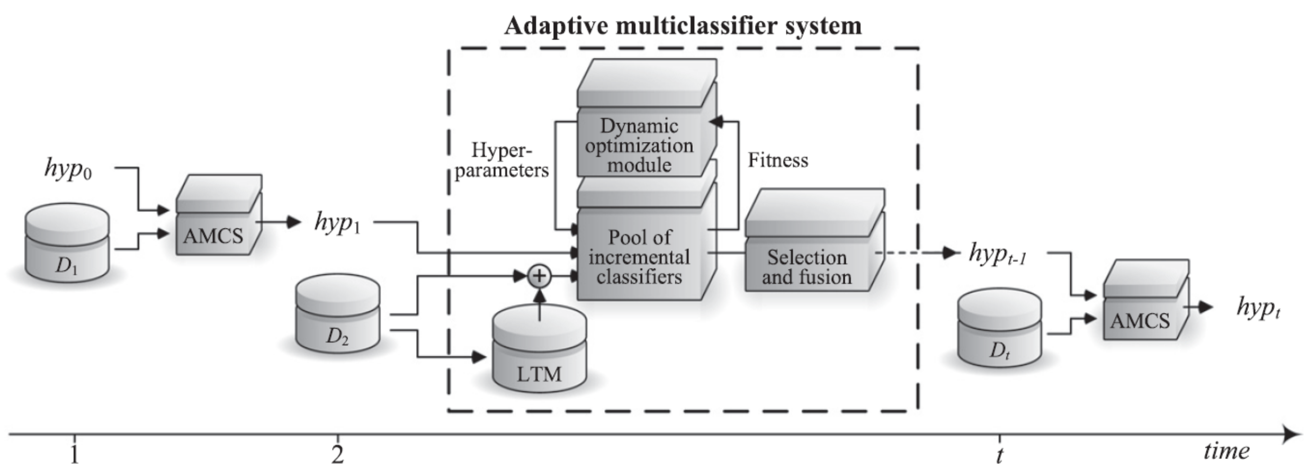


Fig. 2. Evoluti... of classifiers. hyp_0 accordin... ...to update a swarm ...itial hypothesis

Long Term Memory: stores reference samples for validation

Swarm (Pool) of Incremental Learning Classifiers: guided by a dynamic PSO algorithm

EoC Selection and Fusion of base classifiers from the swarm

D.2(4) Optimisation du FAM

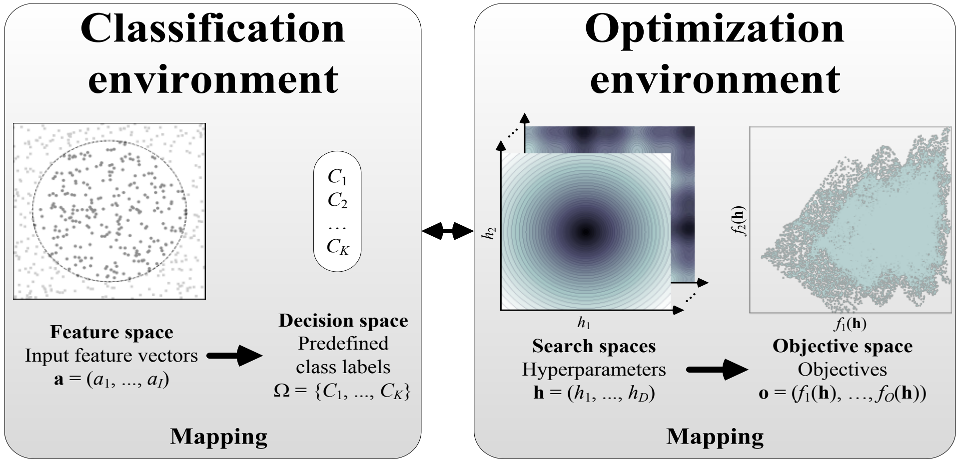
- **DPSO Strategy for Incremental Learning – given a new block of data D_t, evolve an EoFAMs as follows:**

Generation of diversified pools	- DNPSO algorithm to re-optimize parameters of N classifier such that f(**h**,t) is maximized
↓	
Selection of base classifiers	- direct selection of classifiers using particle swarm properties (optima: local best particles)
↓	
Fusion of classifiers	- basic majority voting for output predictions

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Dynamic PSO (DPSO) algorithms allow to:**
  - maintain diversity in the **h** space
  - detect and track several optima in **h** space over time
- **Since h governs learning dynamics,** diversity among particles also assures diversity among corresponding classifiers



D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Dynamic Niching PSO [Nickabadi, 2008]:**
  - **maintains diversity *among* subswarms**
  - **neighborhood topology:** dynamically create subswarms around *local best* particle positions
    - the size of neighborhoods is defined by the distance among particles
  - **free particles (not in a subswarm):** explore independently
    - re-initialized if they converge to non-optimal solutions

(3) DPSO Strategy for Incr. Learning

Ensembles évolutives avec DPSO

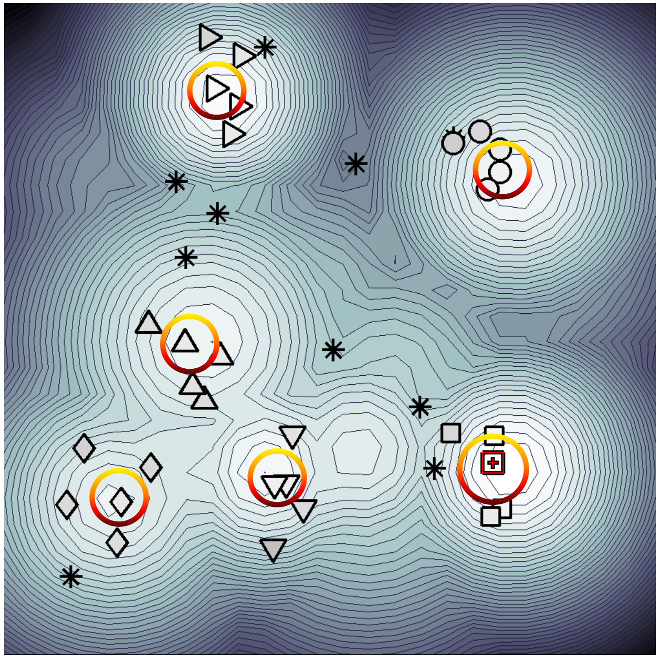
- **Modified Speciation PSO [Blackwell, 2008)]:**
  - **maintains diversity *among and within* subswarms**
  - **neighborhood topology:** groups subswarms around *local best* particle positions
    - ranks particle by fitness
    - **lbest** is defined as particle with best fitness outside range of other subswarms
  - **anti-convergence:** re-initialized particles from least fit subswarm
  - **quantum cloud re-sampling procedure:** randomly repositions particles around center of their subswarm

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Particle properties are used for direct selection**

1. **Initial selection:** classifiers associated with positions of the *local best particles*
2. **Second selection:** greedy search among remaining particles that seeks to increase the average diversity among *particles*



Low cost: avoids computing classifier diversity indicators in input features space

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

NRC-IIT data [Gorodnichy, NRC-48216, 2005]

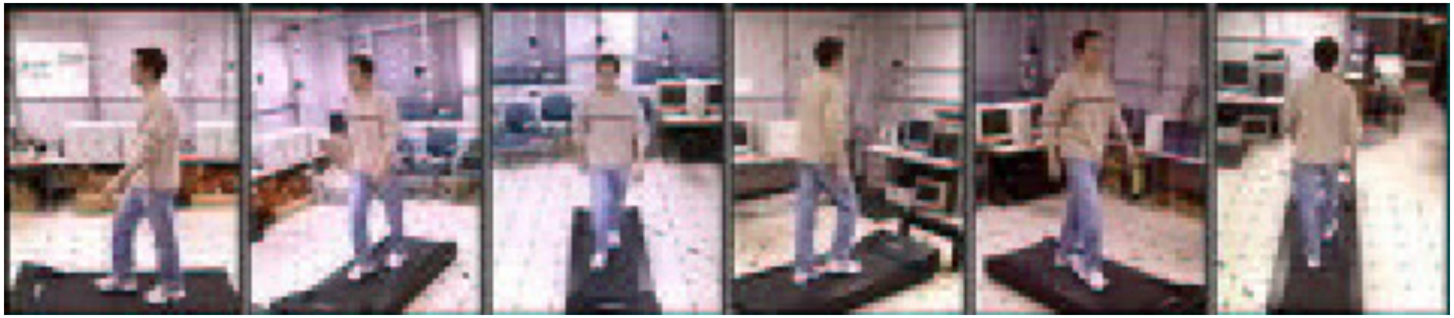
- **Task:** bio-login, recognize the user of a PC
- **11 individuals:** 2 video sequences per individual, one dedicated for training and the other for testing
- **challenging conditions:** changes in pose, expression and proximity, motion blur, low resolution and partial occlusion

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

CMU-MoBo data [Gross et al., CMU-RI-TR-01-18, 2001]

- **Task:** recognize subjects walking
- **25 individuals:** several indoor videos per individual, 6 cameras and 4 motions
- **challenging conditions:** changes in pose, expression, blur, low resolution and partial occlusion



D.2(4) Optimisation du FAM

Evolving Ensembles with DPSO

- **Performance for the update scenarios:**
  [Connolly *et al.*, PR 2012]

Table 4
Average classification rate (in percentage), compression and ensemble size after incremental learning of all the IIT-NRC and MoBo data bases for the update scenario. Each cell is presented with the 90% confidence interval.

Type of learning method	Incremental			Batch	
	LBESTS$_{+d}$	SWARM	GBEST	PSO$_B$	*k*NN
IIT-NRC data base					
Classification rate (%)	82.3 ± 0.4	82.8 ± 0.4	74.9 ± 0.6	82.7 ± 0.2	80.9 ± 0.3
Compression	0.38 ± 0.03	0.13 ± 0.01	8 ± 2	0.062 ± 0.003	1 ± 0
Ensemble size	12.8 ± 0.6	40 ± 0	1 ± 0	40 ± 0	1 ± 0
MoBo data base					
Classification rate (%)	92 ± 2	91 ± 5	89.2 ± 0.7	94.9 ± 0.1	94.5 ± 0.1
Compression	1.3 ± 0.1	0.48 ± 0.02	9.0 ± 0.9	0.09 ± 0.02	1 ± 0
Ensemble size	12.1 ± 0.4	40 ± 0	1 ± 0	40 ± 0	1 ± 0

D.2(4) Optimisation du FAM

Ensembles évolutives avec DPSO

- **Cumulative Match Curves:**

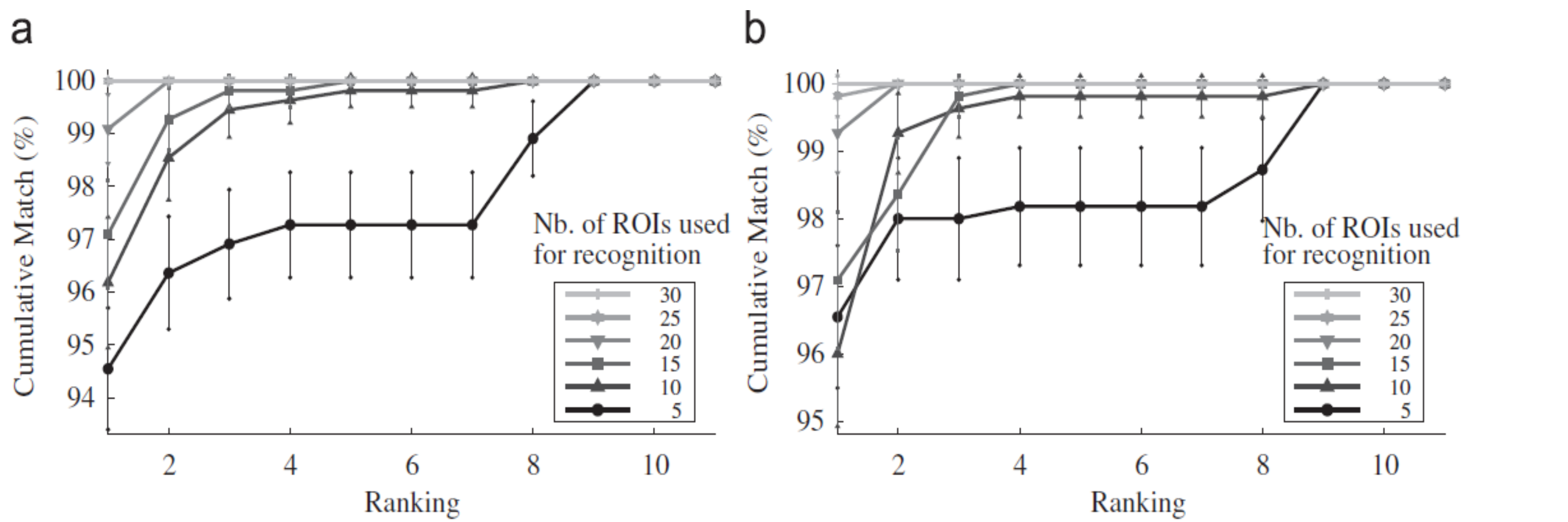


Fig. 11. Cumulative match curves the AMCS's ensemble for different number of ROIs used to perform face recognition. Performance is shown after incremental learning of all the IIT-NRC data base, under both scenarios for the AMCS with LBESTS_{+d}. Error bars correspond to the 90% confidence interval. (a) Enrollment scenario, (b) update scenario.

D.2(4) Optimisation du FAM

Stratégie d'appentissage MOPSO

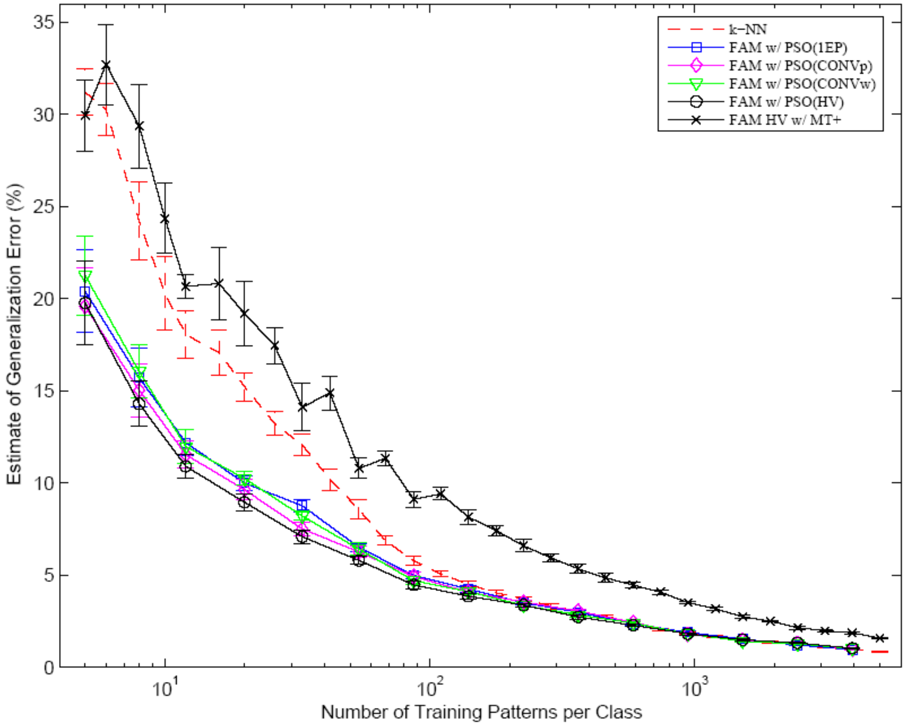
[Prieur et al., CEC2010]

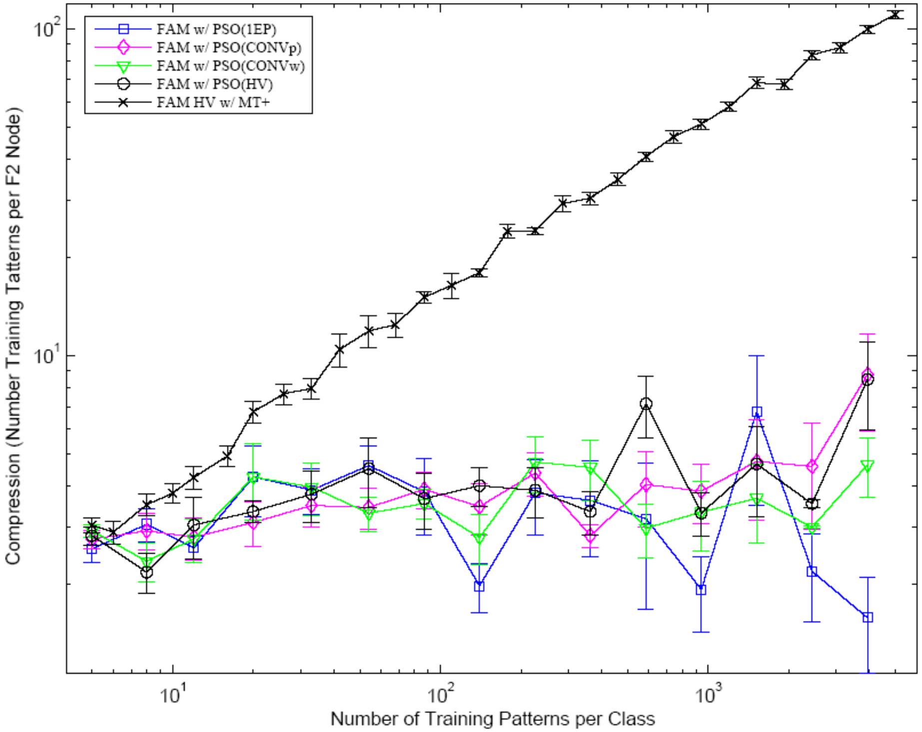
- **Determines FAM weights + architecture + parameters such that the *error rate* and *resources* are minimized:**
  - inspired by an approach by Coello Coello *et al.* (2004) – the leader selection technique is based on Pareto dominance
  - **external archive:** the algorithms outputs a set of non-dominated FAM networks (solutions on the Pareto front)
  - allows to trade-off accuracy *vs.* resources, and discover more cost-effective FAM networks

D.2(4) Optimisation du FAM

Stratégie d'appentissage MOPSO

- $D_{\mathbf{cis}}$ data: average E_{gen} *vs.* training set size (Granger *et al.*, JPRR 2007)





D.2(4) Optimisation du FAM

Stratégie d'appentissage MOPSO

A. Initialization:

- set MOPSO parameters and counters
- initialize particle positions

B. Iterations:

while $q \leq q_{\max}$ iterations do

1. for $i = 1, 2, \ldots, P$ particles
   - train FAM network using parameters of $\mathbf{h}_i^q$
   - compute fitness value $F(\mathbf{h}_i^q)$ of network on validation data
   - update particle *pbest* $\mathbf{p}_i$ if $F(\mathbf{h}_i^q)$ dominates $F(\mathbf{p}_i)$
   - management (add/remove) of leader particles in archive
2. for $i = 1, 2, \ldots, P$ particles
   - select a leader particle from archive
   - compute particle velocity $\mathbf{v}_i^{q+1}$ and update its position $\mathbf{h}_i^{q+1}$
3. increment iteration counter $q = q + 1$

D.2(4) Optimisation du FAM

Stratégie d'appentissage MOPSO

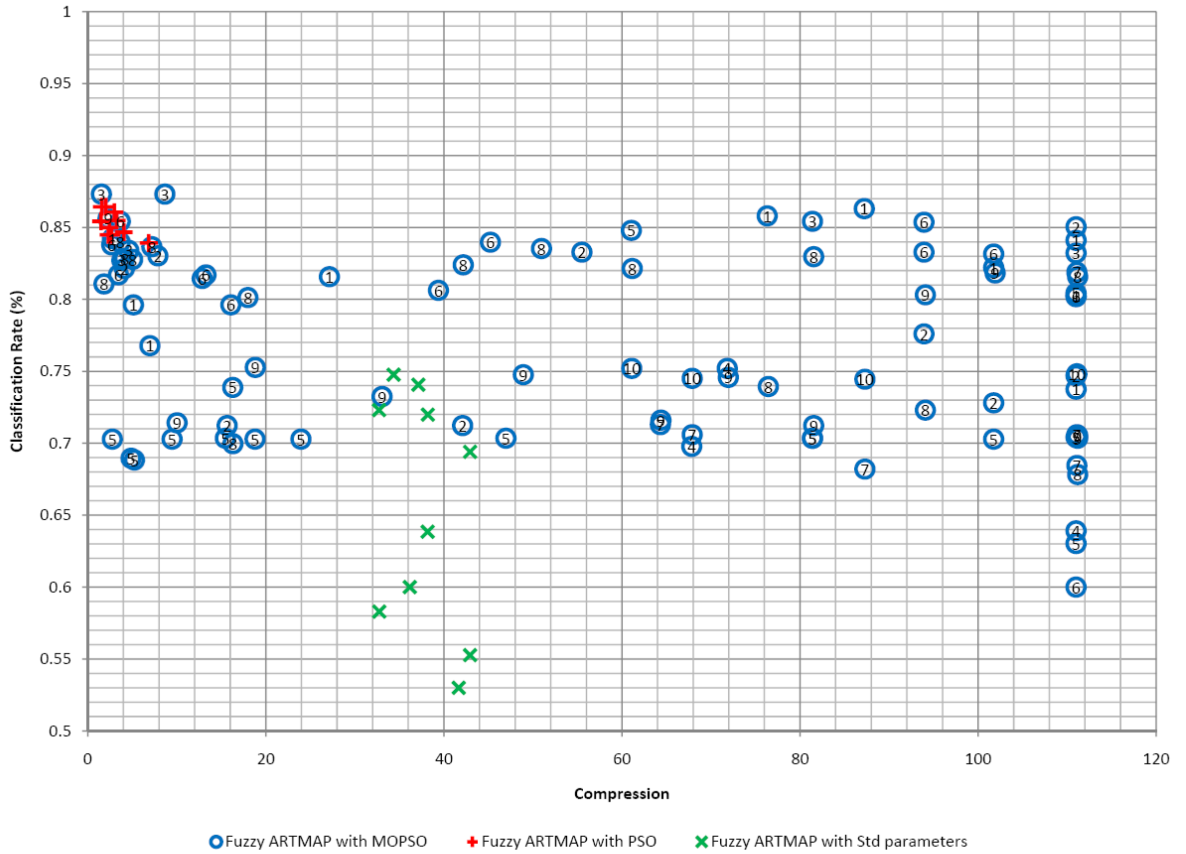
- **Protocol: *k*-fold cross validation:**
  - divide data for learning into $k = 10$ folds
  - fixed parameters: $\beta = 1,\ \alpha = 0.001,\ \bar{\rho} = 0,\ \varepsilon = 0.001$
  - stop training epochs: $CR(\texttt{VAL}_1)_e - CR(\texttt{VAL}_1)_{e\text{-}1} < 0.001$

- **Trials with the PSO/MOPSO strategies:**
  - **each replication is performed with P = 32 particles**
    - particle positions are initialized randomly, except $\mathbf{s}_i^0$ is set to minimize resources
  - **select a second $\texttt{VAL}_2$ set for fitness evaluation**
  - **a trial ends if:** $q_{\max} = 25$ iterations

(4) Simulation Results

Stratégie d'appentissage MOPSO

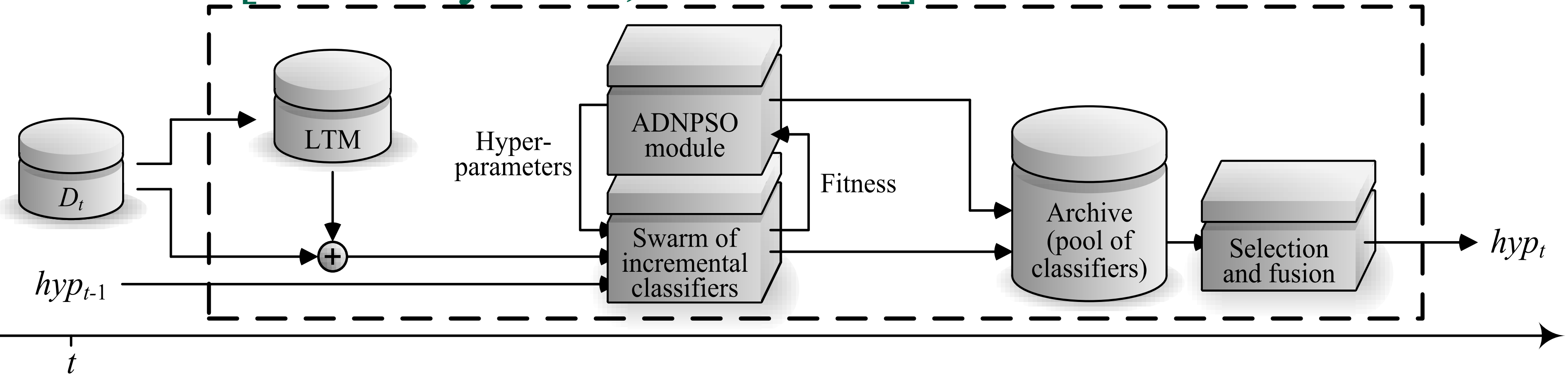
- **Batch learning of NRC data:** error rate *vs.* compression

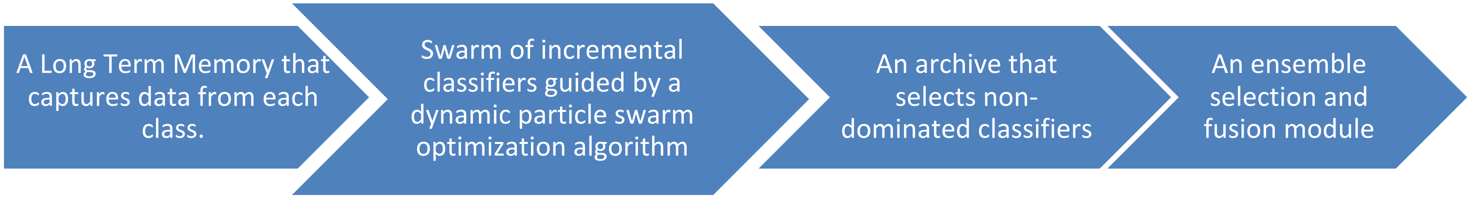


D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

- **ADNPSO:** mimetic approach for multi-objective optimization (to minimize error rate and complexity) [Connolly et al., ASC2013]

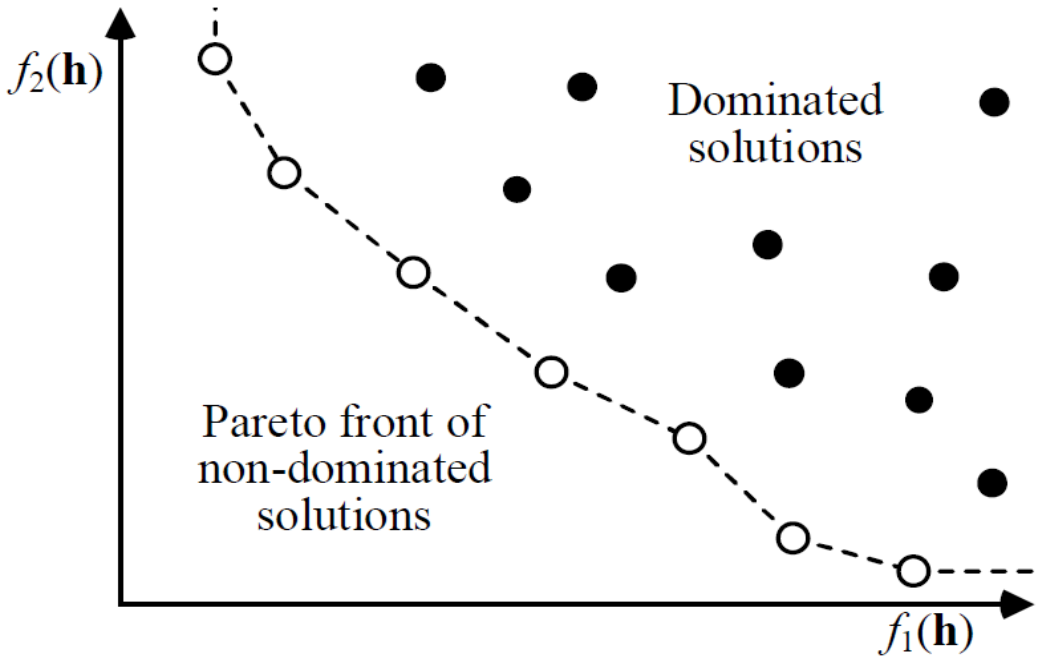




D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

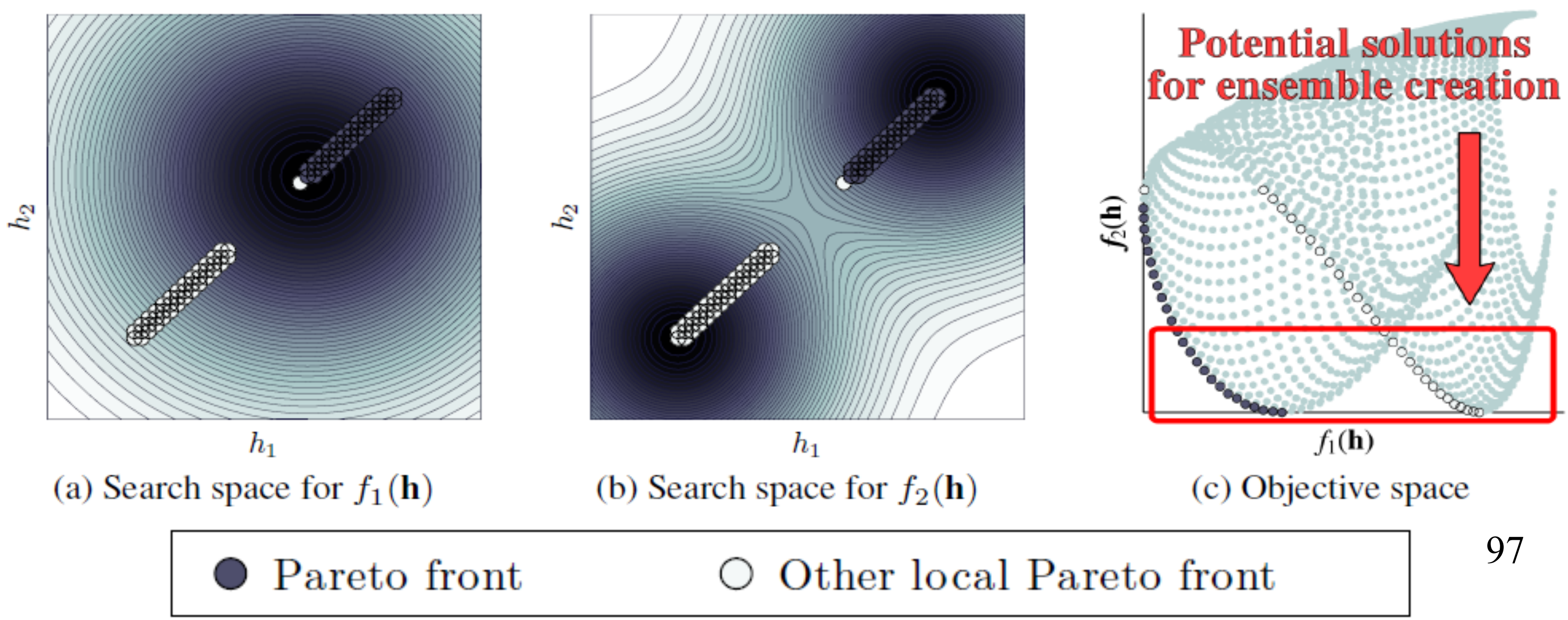
- **ADNPSO:** mimetic approach for multi-objective optimization (to minimize error rate and complexity)



D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

- **Position of local Pareto fronts for 2 search spaces:**
  - **MOPSO:** seeks to find true optimal Pareto-optimal solutions [dark]
  - **ADNPSO:** seeks to search both spaces to find diverse (locally Pareto-optimal) solutions [light]
  - detect local Pareto front to find solutions between the local optima



D.2(4) Optimisation du FAM

Equation to define PSO motion

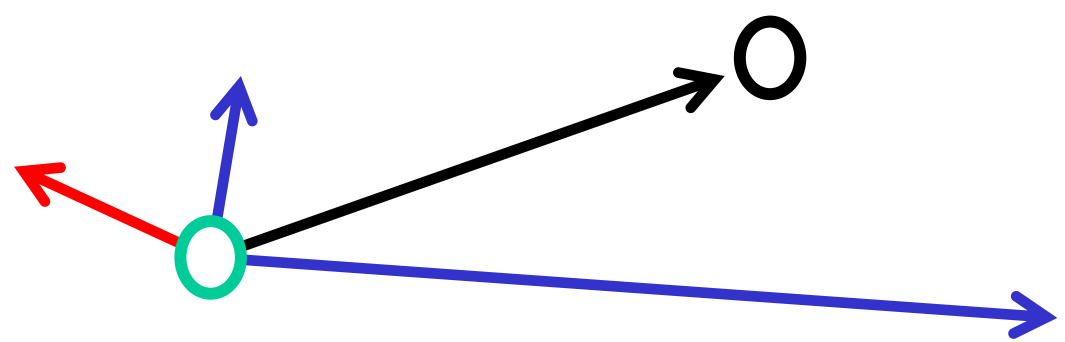
Final position **Inertia**

$$\boxed{\mathbf{h}(\tau+1)} = \boxed{\mathbf{h}(\tau)} + \boxed{w_0\left(\mathbf{h}(\tau) - \mathbf{h}(\tau-1)\right)} + \boxed{\sum_{\phi=1}^{\Phi} r_\phi\, w_\phi \left(\mathbf{h}_\phi - \mathbf{h}(\tau)\right)}$$

Initial position

A sum of influences

Illustration



D.2(4) Optimisation du FAM

Equation to define ADNPSO motion

ADNPSO: Aggregated dynamical niching PSO

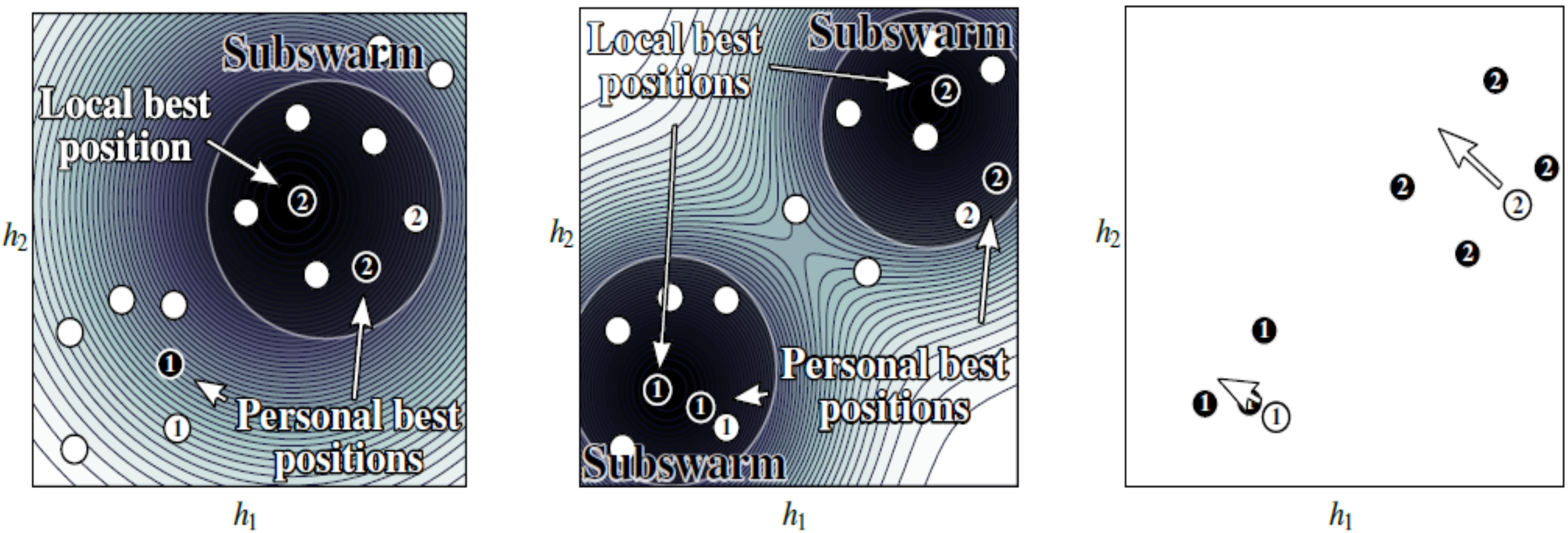
- Cognitive influence: personal best position
- Social influence: local best position

$$\boxed{\mathbf{h}\left(\tau+1\right)} = \boxed{\mathbf{h}(\tau)} + \boxed{w_0\,\left(\mathbf{h}(\tau)-\mathbf{h}(\tau-1)\right)}$$
$$\begin{aligned}+\,& r_1\; w_1\; \left(\mathbf{h}_{\text{social influence, error rate}} - \mathbf{h}(\tau)\right)\\ +\,& r_2\; w_2\; \left(\mathbf{h}_{\text{cognitive influence, error rate}} - \mathbf{h}(\tau)\right)\\ +\,& r_3\; w_3\; \left(\mathbf{h}_{\text{social influence, network size}} - \mathbf{h}(\tau)\right)\\ +\,& r_4\; w_4\; \left(\mathbf{h}_{\text{cognitive influence, network size}} - \mathbf{h}(\tau)\right)\end{aligned}$$

D.2(4) Optimisation du FAM

With sub-swarms defined dynamically!

○ **Particles** ● **Influences**

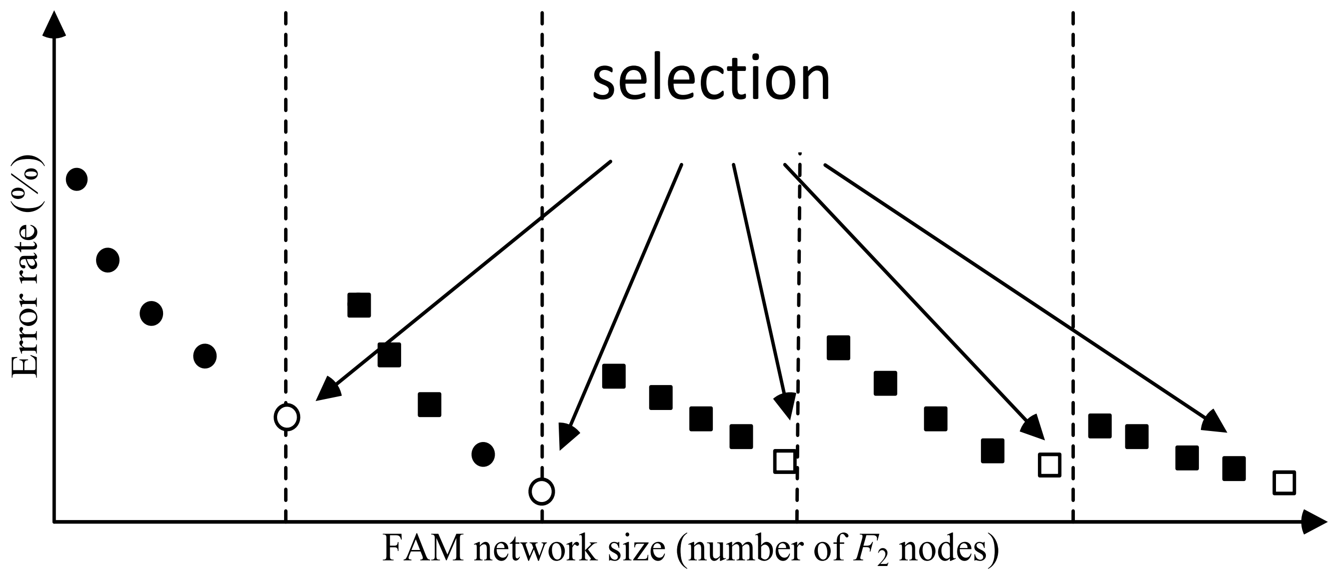


(a) Search space for $f_1(\mathbf{h})$ (b) Search space for $f_2(\mathbf{h})$ (c) Particle movements

D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

- **R**equires a specialized archive and ensemble selection



Selection:

1. most accurate FAMs per network size domain (phenotype diversity)
2. additional greedy search selection to maximize genotype diversity

D.2(4) Optimisation du FAM

Protocol

- **Performance is assessed for AMCSs designed and update according to 3 different IL strategies:**
  1. ***DNPSO:*** Dynamic Niching PSO
     selects local best DNPSO particles + greedy search in swarm
  2. ***MOPSO:*** Multi-Objective PSO
     uses archive and notion of dominance to guide particles towards Pareto-optimal front
  3. ***ADNPSO:*** Aggregated DNPSO
     uses archive with phenotype local best particles + greedy search to maximize genotype diversity

D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

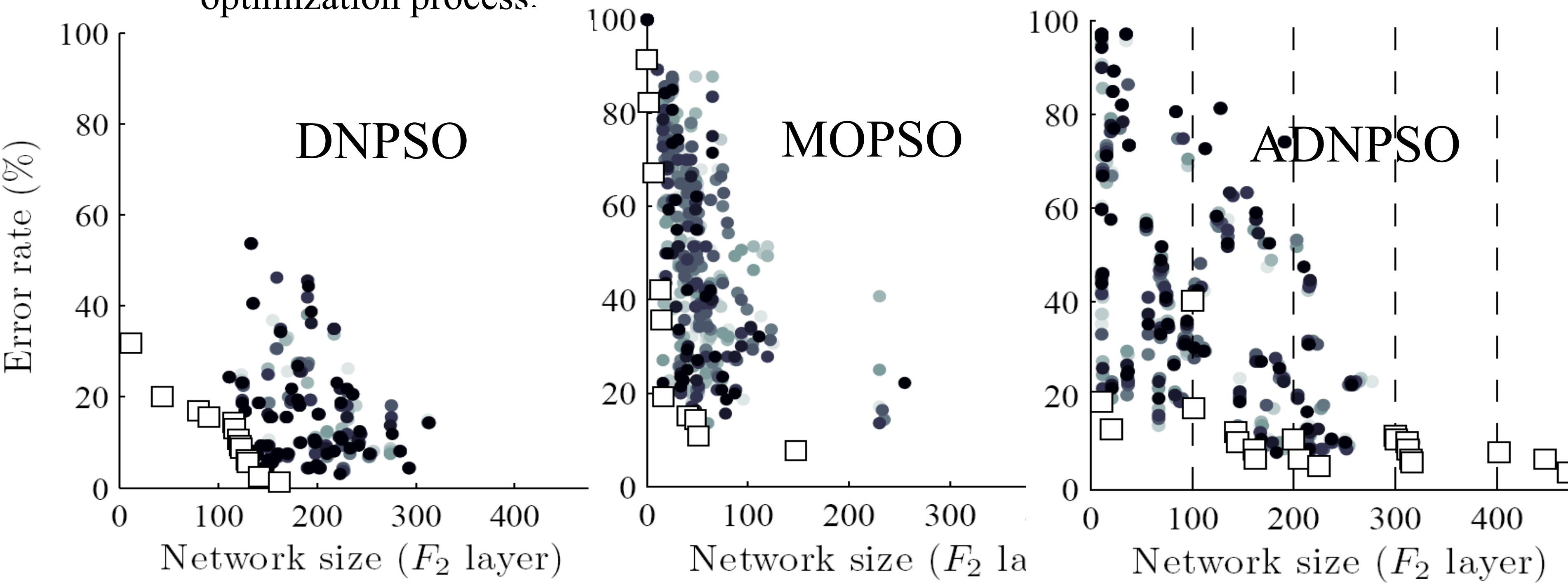
- **Error rate and complexity indicators of approaches:**
  - after incremental learning of all 12 blocks
  - **complexity:** ensemble size, average and total number of *F2* nodes for entire ensemble

Method	**ADNPSO**	DNPSO	MOPSO
Error rate (%)	22.4±0.6	22.7±0.7	26.9±0.7
Ensemble size	5.5 ± 0.4	12.4 ± 0.8	7.9 ± 0.6
Av. nb. of F_2 nodes	170±9	108±5	52±3
Tot. nb. of F_2 nodes	900±100	1300±100	420±50

D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

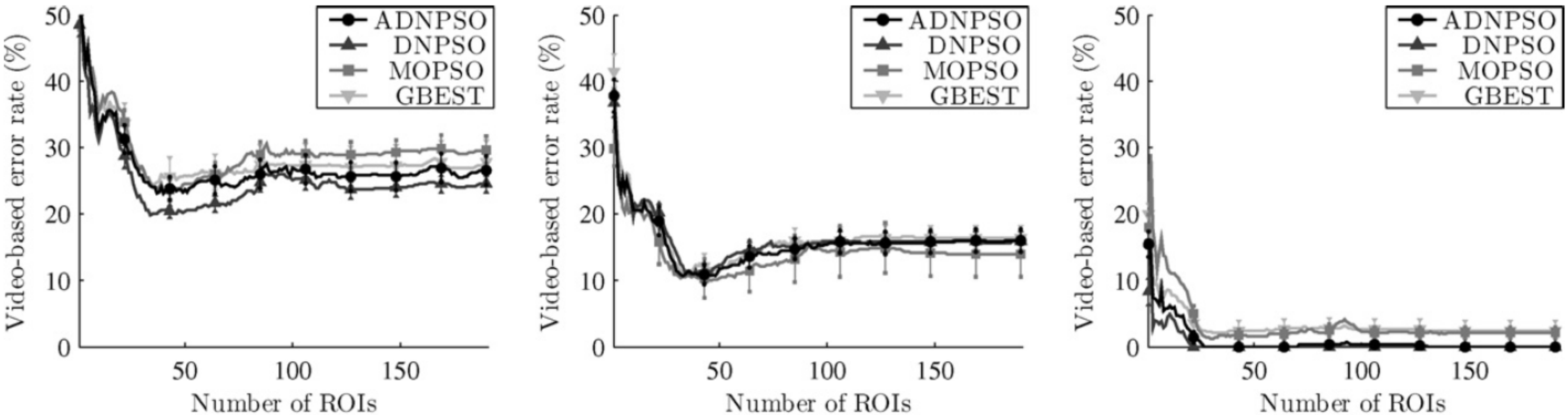
- **Examples of results in the objective space**
  - **circles:** show evolution of the swarm after learning of all 12 blocks
  - **light and dark circles**: position of each particle at the start and end of optimization process.



D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

- **Error rates versus the number of ROIs used to identify individuals over time:**
  - in video surveillance, predictions are accumulated, and several predictions are used for FR

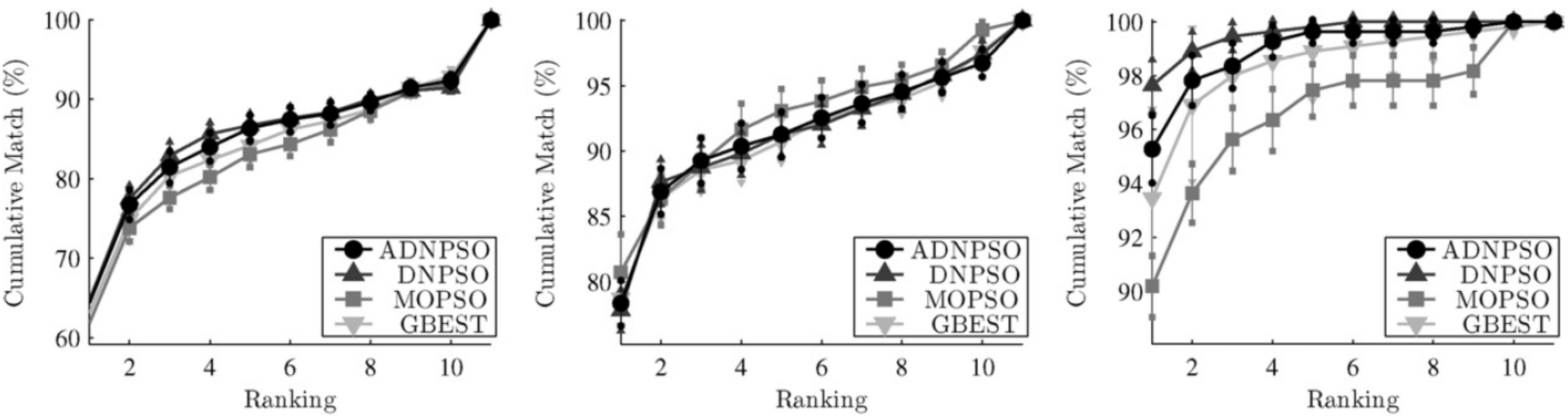


(a) $t = 1$ (b) $t = 6$ (c) $t = 12$

D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

- **Cumulative match curves over time:**
  - in video surveillance, predictions are accumulated, and several predictions are used for FR
  - performance when 15 ROIs are used to perform recognition



(a) $t = 1$ (b) $t = 6$ (c) $t = 12$

D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

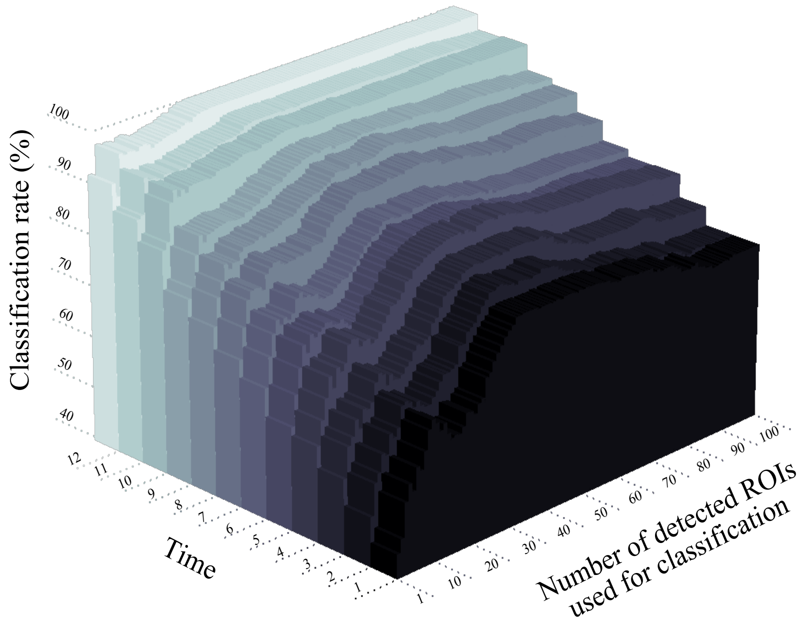
- Incremental learning strategy based on ADNPSO allos to evolve EoCs in response to new data
  - **ADNPSO algorithms:** allows multi-objective min, where both genotype and phenotype diversity are maintained
  - **specialized archive:** local Pareto-optimal solutions found by ADNPSO can be stored and combined
  - EoCs based on accuracy + genotype and phenotype diversity

- **Results:** ADNPSO yields a high level of accuracy comparable to mono-objective DNPSO (higher than MOPSO), but with a fraction of the complexity

D.2(4) Optimisation du FAM

Ensembles évolutives à Critères Multiples

- **Impact of incremental learning on video-based classification rate:**
  - EoCs formed using DNPSO on IIT-NRC update scenario



D.2(4) Optimisation du FAM

- **Accuracy:** number of regions of interest to reach an error rate comparable to 0%

Type of learning	Incremental				Batch	
Method	ADNPSO	DNPSO	MOPSO	GBEST	PSO_B	kNN
IIT-NRC data base						
Enrollment learning scenario						
Minimal av. error rate	0.6 ± 0.7	0 ± 0	5 ± 3	2.1 ± 1	0 ± 0	0 ± 0
Nb. of ROIs to reach 0%	27	22	never	never	20	23
Update learning scenario						
Minimal av. error rate	0 ± 0	0 ± 0	1.2 ± 0.6	0.8 ± 0.5	0 ± 0	0 ± 0
Nb. of ROIs to reach 0%	31	20	never	never	20	23
MoBo data base						
Enrollment learning scenario						
Minimal av. error rate	0 ± 0	0 ± 0	0.5 ± 0.2	1.2 ± 1.4	0 ± 0	0 ± 0
Nb. of ROIs to reach 0%	30	28	never	25	30	16
Update learning scenario						
Minimal av. error rate	0 ± 0	0 ± 0	3 ± 1	0.3 ± 0.3	0 ± 0	0 ± 0
Nb. of ROIs to reach 0%	27	24	never	25	30	16

D.2(4) Optimisation du FAM

▶ Computational complexity: network size

Type of learning Method	Incremental **ADNPSO**	 DNPSO	Batch $\mathrm{PSO_B}$	 *k*NN
IIT-NRC data base				
Enrollment learning scenario				
Ensemble size (↓)	4.5 ± 0.4	19.4 ± 0.7	60 ± 0	**1 ± 0**
Average comp. (↑)	**9.3 ± 0.7**	6.7 ± 0.3	2.2 ± 0.2	1 ± 0
Total comp. (↑)	**2.1 ± 0.2**	0.34 ± 0.02	0.037 ± 0.003	1 ± 0
Update learning scenario				
Ensemble size (↓)	5.5 ± 0.4	19.5 ± 0.7	60 ± 0	**1 ± 0**
Average comp. (↑)	**7.4 ± 0.4**	5.8 ± 0.2	2.2 ± 0.2	1 ± 0
Total comp. (↑)	**1.4 ± 0.2**	0.30 ± 0.03	0.037 ± 0.003	1 ± 0
MoBo data base				
Enrollment learning scenario				
Ensemble size (↓)	7.5 ± 0.5	23.3 ± 0.7	60 ± 0	**1 ± 0**
Average comp. (↑)	**50 ± 6**	23 ± 2	3.6 ± 0.1	1 ± 0
Total comp. (↑)	**6.7 ± 0.7**	1.0 ± 0.1	0.060 ± 0.004	1 ± 0
Update learning scenario				
Ensemble size (↓)	5.5 ± 0.8	19.4 ± 0.8	60 ± 0	**1 ± 0**
Average comp. (↑)	**28 ± 1**	18.6 ± 0.8	3.6 ± 0.1	1 ± 0
Total comp. (↑)	**5.1 ± 0.5**	0.9 ± 0.1	0.060 ± 0.004	1 ± 0