



**HAL**  
open science

## Reconnaissance de Classes Dynamiques : une Modélisation par Réseaux de Petri Flous.

Veyis Gunes, Pierre Loonis, Michel Menard

► **To cite this version:**

Veyis Gunes, Pierre Loonis, Michel Menard. Reconnaissance de Classes Dynamiques : une Modélisation par Réseaux de Petri Flous.. LFA'2001 : rencontres francophones sur la logique floue et ses applications., Nov 2001, Mons, Belgique. hal-02092932

**HAL Id: hal-02092932**

**<https://hal.science/hal-02092932>**

Submitted on 8 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Reconnaissance de Classes Dynamiques : une Modélisation par Réseau de Petri Flou

## Dynamic Classes Recognition : a Fuzzy Petri Net Modelling

V. Gunes

P. Loonis

M. Ménard

Laboratoire d'Informatique et d'Imagerie Industrielles

Université de La Rochelle  
Avenue Michel Crépeau  
17 042 La Rochelle Cedex 1  
vgunes@univ-lr.fr

### Résumé

*Dans cet article, nous étudions les classes d'objets dont les caractéristiques évoluent au cours du temps. La reconnaissance des formes (RdF) d'objets naturels évolutifs fait intervenir ce type de classes dont la modélisation est difficile. Ce problème peut être contourné en rendant dynamique le système de reconnaissance qui prend alors différents états en fonction de l'évolution des classes. Pour ce type d'approche, nous proposons un système de RdF fondé sur l'adaptation d'un ensemble de classifieurs à l'évolution des classes, par l'apprentissage de la planification des différents états à l'aide d'un réseau de Petri flou.*

### Mots Clef

Reconnaissance des Formes, réseau de Petri flou, représentation de connaissance dynamique.

### Abstract

*In this article, we study classes of objects whose features evolve during time. With evolutionary natural objects, the modeling, which involves such classes, is difficult. This problem can be avoided by making dynamic the system of pattern recognition which, then, enters into various states according to the evolution of the classes. For such an approach, we propose a pattern recognition system based on the adaptation of a set of classifiers to the evolution of the classes using a learning stage of the planing by a fuzzy Petri net.*

### Keywords

Pattern recognition, fuzzy Petri net, dynamic knowledge representation.

## 1 Introduction

Lorsque, pour une application donnée, les caractéristiques des objets à reconnaître ne dépendent pas du temps, les classes associées aux différents objets sont dites statiques. Les méthodes de RdF telles que les *c*-moyennes, les *k* plus proches voisins et bien d'autres peuvent être utilisées. Cependant, de nombreuses applications du monde réel font apparaître un glissement des classes dans l'espace des paramètres (usure des machines-outils, dérive des références des instruments de mesure, évolution de la température ambiante, etc.). Dans ces cas, les observations successives (en particulier, celles qui sont à classer) constituent une série temporelle. Un système peut ainsi passer d'une classe à une autre graduellement. Le diagnostic par RdF à boucle de supervision (Boudaoud et Masson [1]), les modèles de Markov cachés pour le diagnostic (Smyth [2]) ainsi que le pronostic par filtrage de Kalman (Frelicot [3]) traitent de ce type d'applications. D'autres applications sont, par nature, dépendantes du temps. Ainsi, les objets naturels sont-ils souvent tributaires d'une évolution, elle même assujettie à des variations périodiques (saisonnnières ou journalières). Nous nous intéressons plus particulièrement à cette seconde catégorie d'applications, dans lesquelles les observations ne constituent plus une série temporelle.

Le problème étudié dans cet article est la détection puis la reconnaissance des différents états d'un système de reconnaissance pour les applications où les caractéristiques des objets à reconnaître subissent des évolutions périodiques (par exemple, saisonnières). Les classes ayant de telles caractéristiques sont des classes

dynamiques appelées aussi classes évolutives. Pour les objets naturels, ces évolutions peuvent être difficilement modélisables par des outils mathématiques. Par conséquent, nous construisons un système de reconnaissance n'associant pas de modèles à la trajectoire des classes.

Les réseaux de Petri (RPs) sont des modèles de représentation bien adaptés à la modélisation des systèmes dynamiques comportant des éléments travaillant en parallèle et de manière asynchrone. Lorsque l'événement du système dynamique est discret, le système est dit Système Dynamique à Événement Discret (SDED) (Cardoso et al. [4]). Les RPs sont couramment utilisés pour modéliser un SDED, chaque état du système étant représenté par un marquage du réseau associé. Cette modélisation présente pourtant l'inconvénient de considérer que le passage d'un état à un autre est un événement discret. Or, dans notre cas, ce passage est continu.

En combinant un tel modèle avec la théorie des ensembles flous, la prise en compte des informations exogènes (connaissances de l'expert, connaissances a priori) et de l'incomplétude de l'information est explicite. En effet, on peut être plus ou moins certain d'être dans un état donné (jeton flou) ou être plus ou moins certain qu'un événement se soit produit (transition floue). Ce type de réseau est connu, dans la littérature, sous le nom de réseau de Petri flou (RPF) et a été proposé pour la première fois par Looney [5]. Une telle approche convient très bien pour le cas où un état du système modélisé correspond à un marquage du RPF associé. Or, un système de reconnaissance dynamique nécessite un RPF dans lequel un état du système correspond à une place (voir la définition de ce mot dans la section suivante) du RPF, le changement d'état se décidant par un système à base de règles floues. Chen et al.[6] proposent également un modèle de RPF, mais les auteurs remarquent que leur modèle ne peut pas être utilisé pour la représentation de connaissance temporelle (Jong et al. [7]). Nous proposons un système de reconnaissance dynamique fondé sur deux types d'apprentissages. L'aspect statique de l'apprentissage est assuré par des classificateurs ou des systèmes de classificateurs (lorsque les classes ont des formes complexes), tandis que l'aspect dynamique est traduit par l'apprentissage de la planification des différents états par un RPF. Il est nécessaire de préciser que notre étude ne concerne pas l'analyse des propriétés du réseau, mais la modélisation d'un système de reconnaissance dynamique par un RPF. Pour une introduction aux propriétés et à l'analyse structurelle des Réseaux de Petri, le lecteur dispose notamment des travaux de Valette et al.[8] et de Murata [9].

La Figure 1 illustre le fonctionnement du système de RdF proposé. Un RPF est construit lors de la phase d'apprentissage en fonction de l'évolution des classes. De même, les transitions  $T_i$  et les jetons des places  $P_i$  sont appris lors de l'apprentissage. Ainsi,  $SC_i$  est spécialisé pour reconnaître les classes  $C_1$  et  $C_2$  pour une partie de leurs trajectoires.

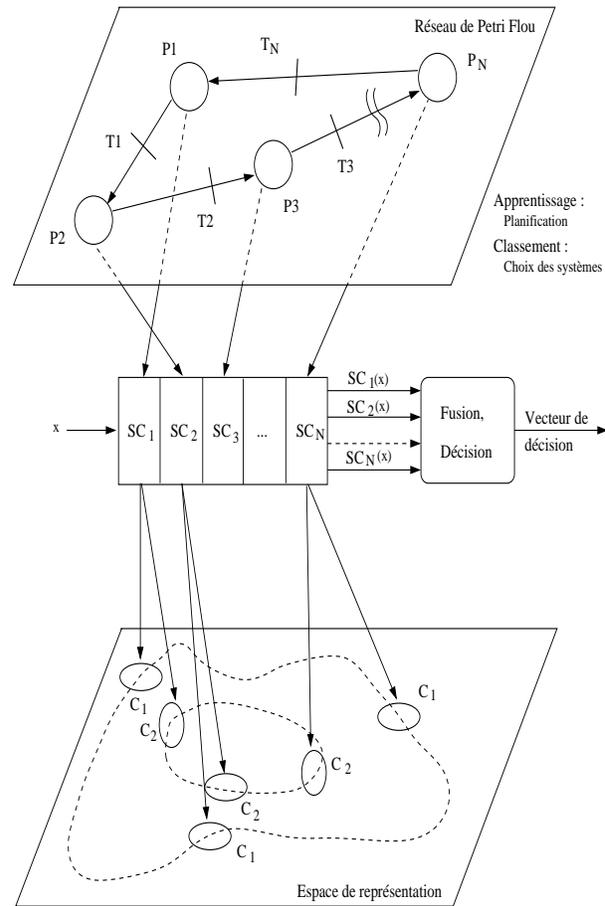


FIG. 1 – Fonctionnement du système de RdF proposé et principe de classement d'un individu inconnu  $x$ . Chaque place ( $P_i$ ) est associée à un état du système de RdF et chaque état du système est représenté par un classifieur ou un Système de Classificateurs ( $SC$ ). L'individu  $x$  est classé par le  $SC$  dont l'état du système est adapté à l'instant considéré. Plusieurs  $SC$ s (les mieux adaptés) peuvent également contribuer à classer l'individu, à travers une phase de fusion et de décision.

Cet article est organisé de la manière suivante. La section 2 traite la représentation de connaissances statiques et présente notre proposition de représentation de connaissances dynamiques par un RPF. Puis, la section 3 détaille la synthèse d'un RPF et définit les jetons, ainsi que les transitions. La section 4 présente

un exemple sur des données synthétiques.

## 2 Représentation de connaissances dynamiques par des réseaux de Petri flous

Dans notre approche, les propositions se réfèrent aux différents états du système de reconnaissance. Ainsi, les changements d'états de ce système sont-ils analogues aux transferts de jetons dans un RP. Il est alors facile d'associer un p-invariant au système de reconnaissance. Un p-invariant est un ensemble de places pour lequel la charge globale en jetons reste constante indépendamment des transitions franchies. Un des avantages de la théorie des réseaux de Petri est que les p-invariants peuvent être calculés par résolution de systèmes linéaires, sans énumération de tous les marquages possibles. Ce type de calcul permet de vérifier que le système de reconnaissance propose, à tout instant, un seul état possible du système. Les différentes places de ce p-invariant représentent alors les différents états du système.

Nous tenons compte du temps explicitement, en définissant les dates floues associées à la fois aux transitions et aux jetons (autant de jetons qu'il y a de places, chaque place ayant un jeton flou qui lui est associé). Ainsi, le temps courant correspond à une date possible pour les franchissements des transitions et les disponibilités des jetons.

### 2.1 RPF pour le raisonnement

En utilisant les RPFs, la règle de production floue  $R_i$  permettant de décrire une relation floue entre deux propositions  $D_j$  et  $D_k$  s'écrit de la manière suivante :

$$R_i : \text{SI } D_j \text{ ALORS } D_k (FC = \mu_i) \quad (1)$$

où  $\mu_i$  est la valeur du Facteur de Certitude  $FC$  (Chen et al. [6]) associé à la transition. Ce facteur qualifie la véracité de la règle (et non celle de la proposition  $D_k$ ). Lorsque au moins deux propositions sont concernées, la règle contient des connecteurs logiques "ET" ou "OU". Ces règles, appelées règles de production floues composites, sont de 4 types (Looney et Alfize [10]) :

- Type 1 : SI  $D_{j1}$  ET ... ET  $D_{jn}$  ALORS  $D_k$  ( $CF = \mu_i$ )
- Type 2 : SI  $D_j$  ALORS  $D_{k1}$  ET ... ET  $D_{kn}$  ( $CF = \mu_i$ )
- Type 3 : SI  $D_{j1}$  OU ... OU  $D_{jn}$  ALORS  $D_k$  ( $CF = \mu_i$ )
- Type 4 : SI  $D_j$  ALORS  $D_{k1}$  OU ... OU  $D_{kn}$  ( $CF = \mu_i$ )

Dans (Jong et al. [7]), les auteurs remarquent que ce modèle ne peut pas être utilisé pour la représentation de connaissance temporelle. Dans leur approche, la partie condition (proposition  $D_j$ ) d'une règle est consommée lorsque la règle est appliquée, de la même manière que le jeton de la place d'entrée est consommé par le franchissement de la transition. Ceci oblige à

considérer qu'une proposition reste vraie uniquement si elle n'est pas utilisée (consommée). Le jeton étant flou, la décision de consommer (i.e., de franchir une transition) dépend du seuil associé au jeton de la place d'entrée. Dans ce cas, on ne peut prétendre que le franchissement d'une transition est flou. Looney [5] utilise également un seuil associé à chaque transition pour décider de transmettre le jeton flou. Il propose d'en transmettre une copie à la place de sortie, pour que la proposition associée à la place d'entrée reste vraie. Ceci se traduit par une boucle simple dans le graphe du réseau de Petri. L'inconvénient est qu'il n'apparaît pas dans la matrice d'incidence du réseau. Au lieu de se donner un choix uniquement entre les solutions "copier" ou "consommer" (type binaire : 0 ou 1), il serait plus pertinent d'alimenter correctement toute place avec une charge adéquate d'un jeton associé à cette place. Notre étude montre qu'il est possible de construire des fonctions d'appartenances de tels jetons lors de la phase d'apprentissage (voir la section 3.1).

### 2.2 RPF pour connaissances dynamiques

Dans l'hypothèse des évolutions périodiques déterministes, les parties antécédent et conséquence d'une règle de production floue sont uniques, car les propriétés du temps sous-jacent sont celles de  $\mathbb{N}$  (l'ensemble des entiers naturels) relativement à la relation précédente (ordre total discret) : étant donné un premier état, chaque état suivant induit un successeur unique. Ainsi, nous utilisons uniquement les règles qui sont semblables à celle décrite par  $R_i$  (1). Cependant, le modèle de RPF proposé ci-dessous reste valable pour toutes les autres règles. C'est le cas dans l'hypothèse des problèmes non-déterministes, dans lequel le temps peut avoir plusieurs futurs, même si pour tous les états d'une série donnée, il a un passé unique.

Nous proposons de modifier le modèle de Chen et al. [6] de manière à tenir compte explicitement du temps. Chaque jeton associé à une place est un ensemble flou dont la fonction d'appartenance qui le définit dépend du temps. Chaque transition n'est plus associée à un facteur de certitude (indépendant du temps), mais est un ensemble flou dont la fonction d'appartenance qui le définit dépend du temps. Notre RPF n'est pas asynchrone dans le sens où une transition peut être franchie à un instant inconnu (contrairement au réseau de Petri classique), mais il est asynchrone dans le sens où le RPF peut être utilisé à des instants inconnus : c'est l'application des règles qui se produit à des instants indéterminés.

Pour la phase de classement, nous proposons de ré-initialiser le marquage en fonction des valeurs prises

par les jetons à l'instant du classement, avant de faire appel au système de reconnaissance par l'application des règles. Nous proposons de définir un RPF de la manière suivante :

**Definition 1** *Un réseau de Petri flou est un 8-uplet  $RPF = (\mathcal{P}, \mathcal{T}_{rpf}, \mathcal{D}, I, O, f, \alpha, \beta)$  où :*

- $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$ , un ensemble fini de places,
- $\mathcal{T}_{rpf} = \{T_1, T_2, \dots, T_M\}$ , un ensemble fini de transitions,
- $\mathcal{D} = \{D_1, D_2, \dots, D_N\}$ , un ensemble fini de propositions,
- $\mathcal{P} \cap \mathcal{T}_{rpf} \cap \mathcal{D} = \emptyset, |\mathcal{P}| = |\mathcal{D}|$ ,
- $I : \mathcal{T}_{rpf} \rightarrow \mathcal{P}^\infty$  est la fonction d'entrée, qui associe à toute transition un ensemble de places,
- $O : \mathcal{T}_{rpf} \rightarrow \mathcal{P}^\infty$  est la fonction de sortie, qui associe à toute transition un ensemble de places,
- $f : \mathcal{T}_{rpf} \times \mathcal{T} \rightarrow [0, 1]$  est la fonction d'association, qui associe à toute transition un réel compris entre 0 et 1, à tout instant  $t \in \mathcal{T}$ ,
- $\alpha : \mathcal{P} \times \mathcal{T} \rightarrow [0, 1]$  est la fonction d'association, qui associe à toute place un réel compris entre 0 et 1, à tout instant  $t \in \mathcal{T}$ ,
- $\beta : \mathcal{P} \rightarrow \mathcal{D}$  est une fonction d'association bijective entre places et propositions.

La règle de production floue  $R_i$  devient alors :

$$R_i : \text{SI } D_j \text{ ALORS } D_k (FC = \mu_{T_j}(t)) \quad (2)$$

où  $\mu_{T_j}(t) = f(T_j, t)$  est la valeur du Facteur de Certitude. De même,  $\mu_{P_j}(t) = f(P_j, t)$  est la "valeur" (i.e., le degré d'appartenance) du jeton de la place  $P_j$  à l'instant  $t$ .

Si  $\alpha(P_j) = y_j, y_j \in [0, 1]$  et  $\beta(P_j) = D_j$ , alors le degré de vérité de la proposition  $D_j$  est  $y_j$ , i.e., le degré de vérité de la validité de l'état  $j$  du système de reconnaissance est  $y_j$ . Cette règle peut être modélisée comme dans la Figure 2, où  $P_j$  et  $P_k$  sont respectivement places d'entrée et de sortie de la transition  $T_j$ . Le degré de vérité de la proposition  $D_k$  (système de reconnaissance dans l'état  $k$ ) est alors :

$$y_k = y_j \times \mu_{T_j}(t) = 0.8 * 0.9 = 0.72$$

### 3 Synthèse d'un RPF

Un modèle de représentation de connaissances dynamiques étant proposé, les fonctions d'appartenance doivent être déterminées (i.e., apprises) lors de la phase d'apprentissage.

#### 3.1 Aspect dynamique de l'apprentissage

Afin de générer un RPF, il est nécessaire de détecter les changements d'états et de déterminer les fonctions

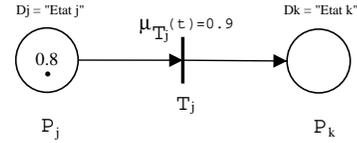


FIG. 2 – Représentation d'une règle de production floue par un RPF.

d'appartenance des transitions et des jetons. Des mesures de changements fondées sur l'organisation des caractéristiques ont été proposées par Sarkar et Boyer [11]. A partir de valeurs et vecteurs propres de la matrice d'adjacences, leur méthode permet de quantifier le changement des données. Ces mesures, associées à des seuils peuvent permettre de créer un nouvel état. L'instant associé à cet événement peut servir au choix de la distribution des appartenances pour la transition d'entrée de la place associée au nouvel état.

Dans le cadre de la RdF, nous proposons une méthode moins coûteuse en temps de calcul, qui consiste à observer l'évolution de la matrice de confusion à partir de l'apprentissage effectué pour une configuration initiale. Une phase d'apprentissage permet de détecter les changements d'états nécessaires pour ne pas dépasser un certain seuil de confusion entre les classes. Les instants associés à ces changements servent à déterminer les dates floues pour les transitions et les jetons associés à chaque place, i.e., de construire leurs fonctions d'appartenance.

L'évolution étant périodique, à tout instant, une seule transition peut être franchie. Ceci implique que l'imprécision temporelle doit être faible, du fait que nous sommes amenés à choisir un seul état à tout instant, bien qu'il soit imaginable de prévoir le cas où l'on déciderait de l'affectation à une classe suite à la fusion des résultats de plusieurs *systèmes* de classifieurs. Un *système* de classifieurs peut être composé de plusieurs classifieurs. De tels *systèmes* s'avèrent utiles lorsque les classes sont multimodales ou lorsqu'elles se chevauchent et nous avons déjà proposé un tel *système* (algorithme 3C [12]). Puisque, à tout moment, le système de reconnaissance peut être, au plus, entre deux états, nous limitons cette étude au cas où seulement deux *systèmes* doivent être choisis.

L'apprentissage permet de construire ces fonctions d'appartenance en tenant compte implicitement des connaissances de l'expert. Chaque instant associé à un changement d'état correspond alors à l'instant auquel le degré d'appartenance de la transition franchie est maximal ( $\mu_{T_n}(t) = 1$ ). Il est raisonnable de considé-

rer que cette fonction d'appartenance doit s'annuler aux instants correspondants aux changements d'états précédent et suivant (i.e., correspondants aux instants pour lesquelles les degrés d'appartenances des transitions précédente et suivante sont maximales). Pour les jetons flous, le raisonnement est un peu différent. En effet, entre deux changements d'états  $n$  et  $n+1$ ,  $\mu_{J_n}(t)$  devrait avoir des degrés d'appartenance élevés et ne devrait pas être nulle à ces deux instants. Car c'est, précisément, à ces deux instants que le système de reconnaissance devra faire un choix permettant d'anticiper sur l'évolution des classes. Par conséquent, nous proposons d'annuler  $\mu_{P_n}(t)$  à des instants avant le changement  $n$  et après le changement  $n+1$ . Un moyen pratique de calculer ces deux écarts respectifs est de diviser par deux les durées de validité des états précédent et suivant. Ainsi, ces deux instants correspondent aux maxima de  $\mu_{P_{n-1}}(t)$  et de  $\mu_{P_{n+1}}(t)$ . De plus, ce choix permet d'avoir un RPF p-invariant. Dans cette étude, nous choisissons des fonctions d'appartenances triangulaires, définies formellement ci-après. Ces deux fonctions d'appartenances sont représentées dans la Figure 3.

**Definition 2** Soit  $t_{T_n} \in \mathcal{T}, \forall n \in [1, N]$ , l'instant correspondant au changement de l'état  $n$  vers l'état  $n+1$ . La fonction d'appartenance  $\mu_{T_n}(t)$  de l'ensemble flou "la transition  $T_n$  est à peu près valide" est définie par :  $\mu_{T_n} : \mathcal{T} \rightarrow [0, 1]$

$$\mu_{T_n}(t) = \begin{cases} \frac{t-t_{T_{n-1}}}{t_{T_n}-t_{T_{n-1}}}, & \text{si } t_{T_{n-1}} \leq t \leq t_{T_n} \\ \frac{-t+t_{T_{n+1}}}{t_{T_n}-t_{T_{n+1}}}, & \text{si } t_{T_n} < t \leq t_{T_{n+1}} \\ 0, & \text{sinon.} \end{cases} \quad (3)$$

**Definition 3** Soit  $t_{T_n} \in \mathcal{T}, \forall n \in [1, N]$ , l'instant correspondant au changement de l'état  $n$  vers l'état  $n+1$  et  $\Delta_n$ , la durée entre  $t_{T_n} \in \mathcal{T}$  et  $t_{T_{n+1}} \in \mathcal{T}$  divisé par 2. La fonction d'appartenance  $\mu_{P_n}(t)$  de l'ensemble flou "le jeton de la place  $P_n$  est à peu près valide" est définie par :  $\mu_{P_n} : \mathcal{T} \rightarrow [0, 1]$

$$\mu_{P_n}(t) = \begin{cases} \frac{t-(t_{T_n}-\Delta_{n-1})}{\Delta_{n-1}+\Delta_n}, & \text{si } (t_{T_n}-\Delta_{n-1}) \leq t \leq (t_{T_n}+\Delta_n) \\ \frac{-t+(t_{T_{n+1}}+\Delta_{n+1})}{\Delta_n+\Delta_{n+1}}, & \text{si } (t_{T_n}+\Delta_n) < t \leq (t_{T_{n+1}}+\Delta_{n+1}) \\ 0, & \text{sinon.} \end{cases} \quad (4)$$

Ainsi, l'état du réseau de Petri flou change-t-il en fonction de l'évolution des transitions floues et de l'évolution des marquages flous associés aux places. De

cette manière, nous prenons explicitement le temps en compte et nous effectuons l'apprentissage dynamique. Ce dernier nous permettra de reconnaître, à tout moment, le "meilleur" *Système* à utiliser. Le degré de véridité d'un état du système de RdF (d'une proposition) est déterminé par la valeur du jeton associé à sa place. Le RPF construit dans ces conditions est forcément vivant et ne comporte, en conséquence, pas de blocages. Il est aussi 1-borné et sauf, par conséquent. Cette dernière propriété implique que le réseau est cohérent, i.e.,  $\forall t \in \mathcal{T}$  et  $\forall n \in [1, N], \mu_{P_n}(t) \leq 1$ . Dans le cas contraire, cela poserait un problème d'interprétation de " $\mu_{P_n}(t) > 1$ ".

### 3.2 Algorithme de construction du système de reconnaissance

L'algorithme suivant permet de déterminer ces instants de changements d'états, instants qui serviront à construire les fonctions d'appartenances. Soit  $n$ , le nombre d'états, égal au nombre de classifieurs (ou de système de classifieurs)

#### Procédure Apprentissage

**Entrée :**  $X^{lr}$  : ensemble d'apprentissage,  $T$ , Tolérance,  $\epsilon$

**Sortie :** Fonctions d'appartenances des jetons et des transitions  $i \in [1, n]$ , les classifieurs ou systèmes  $SC_i$   $i \in [1, n]$

- 1 : Effectuer un apprentissage à  $t = 0$  ( $e_1$ )
- 2 : Effectuer un classement à  $t = 0$  ( $e_1$ )
- 3 : Calculer la matrice de confusion (référence)  $M^0$
- 4 :  $m \leftarrow 0$
- 5 : **répéter**
- 6 :  $n \leftarrow 1; t \leftarrow 0$
- 7 : **répéter**
- 8 : Effectuer un classement (classifieur  $e_n$ )
- 9 : Calculer la matrice de confusion  $M_t$
- 10 : **si** ( $\Delta$  confusion  $>$  Tolérance) **alors**
- 11 :  $t_{T_n} \leftarrow t; n \leftarrow n + 1$
- 12 : Effectuer un apprentissage et un classement ( $e_n$ )
- 13 : Calculer la matrice de confusion  $M^0$
- 14 : **fin si**
- 15 :  $t = t + \delta t$
- 16 : **jusqu'à** ( $t > T$ )
- 17 :  $m_{ancien} \leftarrow m; m \leftarrow n; e \leftarrow e_m$
- 18 : **jusqu'à** ( $(m = m_{ancien}) \text{ ET } (|Centres de classes| < \epsilon)$ )
- 19 : Construire les fonctions d'appartenances

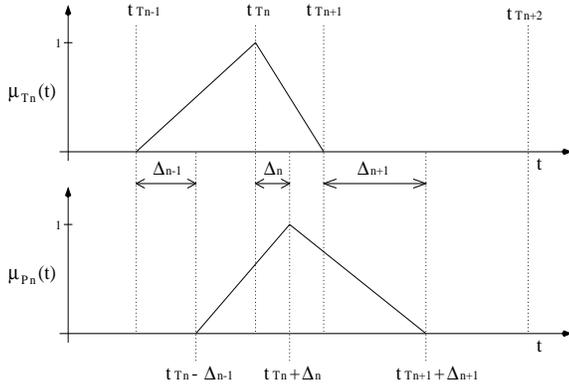


FIG. 3 – Construction des fonctions d'appartenance des transitions  $\mu_{Tn}(t)$  et de celles des jetons  $\mu_{Pn}(t)$ .

## 4 Aspect statique de l'apprentissage

Les connaissances statiques (valables à un moment donné dans une période) sont apprises par les classifieurs ou les systèmes de classifieurs associés aux différents états du système de reconnaissance dynamique. Elles sont modélisées de différents manières (centres de classes, matrices de variance-covariances, poids des entrées de neurones dans un réseau de neurones, etc.), en fonction du type de système de classifieurs.

## 5 Validation

Dans cette étude, chaque système de classifieurs est composé d'un seul classifieur. Ceci est suffisant car les classes ne se chevauchent jamais et restent unimodales. A ce niveau, nous utilisons également les ensembles flous. Une méthode de classification non supervisée, appelée c+2 moyennes flous [13] dont nous avons inhibé les possibilités de rejet de distance et de rejet d'ambiguïté, est utilisée en classement. Ce classement est effectué avec une itération supplémentaire, en utilisant les centres de classes obtenus lors de la phase de coalescence. Il va de soi que d'autres méthodes peuvent également être employées.

### 5.1 Données synthétiques

La Figure 4 présente un problème d'évolution périodique à deux classes dont les trajectoires sont elliptiques, avec croisement entre les deux trajectoires. Ainsi, si l'on prend en compte l'ensemble des individus des classes sur une période, ces classes sont partiellement mélangées. La période est composée de  $n_p = 256$  instants et chaque classe comporte 400 individus. Les confusions et/ou les ambiguïtés entre classes apparaissent dans certaines configurations (si on utilise un

seul classifieur). Pour un problème du monde réel, les trajectoires sont en général plus compliquées, mais peuvent présenter les mêmes caractéristiques (classes nettement séparées). L'évolution de ces classes est définie par les  $n_p$  positions successives occupées, correspondant aux différents instants. Aux instants initial et final, les classes sont générées de la manière suivante (avec l'écart-type  $\sigma = 2$ ) :

–  $\phi_1$  : 400 individus suivant une distribution Gaussienne ;

$$m_1 = ( 30 \ 60 )^t, \Sigma_1 = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

–  $\phi_2$  : 400 individus suivant une distribution Gaussienne ( $\Sigma_2 = \Sigma_1$ ) ;

$$m_2 = ( 90 \ 50 )^t$$

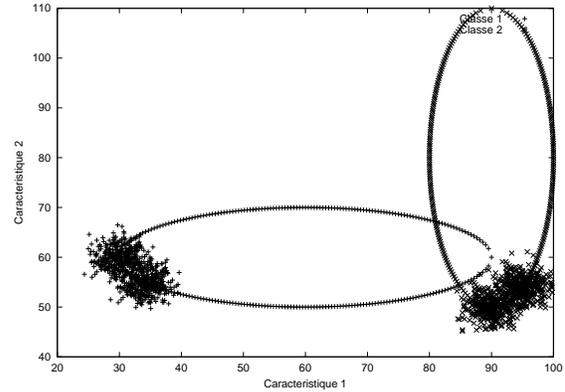


FIG. 4 – Exemple de problème à deux classes de trajectoires elliptiques.

### 5.2 Apprentissage

Après l'apprentissage et le classement (pour le suivi de l'évolution du taux de reconnaissance) dans la position initiale, nous créons un nouvel état à chaque fois que la confusion sur une classe quelconque dépasse un seuil choisi de  $cf_i = 10\%$  de confusion. Lorsque cet événement se produit, le nouvel état est associé à un nouveau classifieur qui apprend sur les données associées à la position concernée et les reclasse. Le classement s'effectue alors avec ce nouveau classifieur jusqu'au prochain dépassement du seuil de confusion. Le cycle se termine lorsque toute la période est ainsi traitée. Soit  $X^{tr}$ , l'ensemble des individus générés dans toutes les positions de la période, qui constitue l'ensemble d'apprentissage. L'ensemble de test  $X^t$  est généré dans les mêmes conditions.

**Fonctionnement.** Si nous réitérons le cycle d'apprentissage plusieurs fois sur une même période, au bout du 3ème cycle, le graphe obtenu devient stable (y

compris les centres de classes associés). Dans ces conditions, nous obtenons l'évolution du taux de reconnaissance (Figure 5). Les centres de classes de chaque état (du système de reconnaissance) sont montrés dans la Figure 6. Les places et transitions successives obtenues sont représentées avec un réseau de Petri dans la Figure 7. Ce résultat diffère si le nombre de positions dans la période ou si le seuil du taux de reconnaissance utilisé est différent. A partir du 2ème cycle, le cycle débute en adoptant (en conservant) le dernier classifieur utilisé lors du cycle précédent. Ceci permet le suivi de l'évolution du taux de reconnaissance.

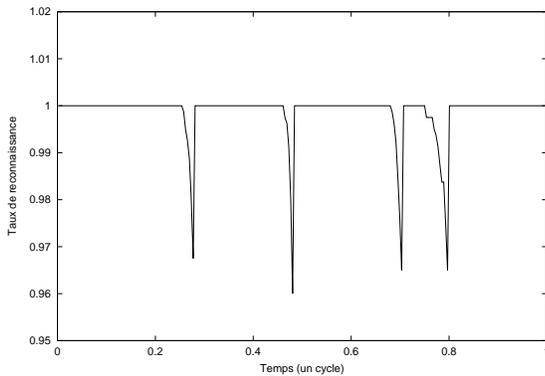


FIG. 5 – Evolution du taux de reconnaissance en tenant compte des changements d'états, lors de l'apprentissage. Les 4 changements d'états nécessaires correspondent aux 4 chutes visibles.

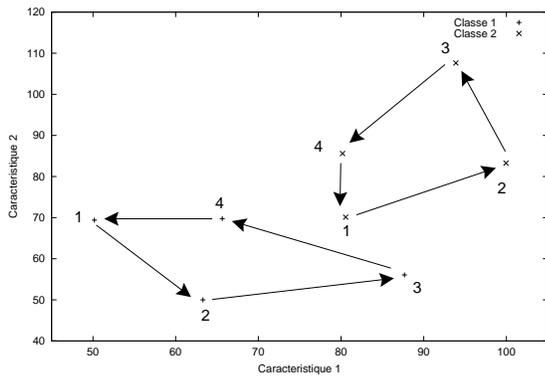


FIG. 6 – Résultat de l'apprentissage des différentes positions des centres de classes (pour les différents états indiqués sur la figure).

**Influence de la position initiale.** Il est évident que le système de reconnaissance obtenu (les états,

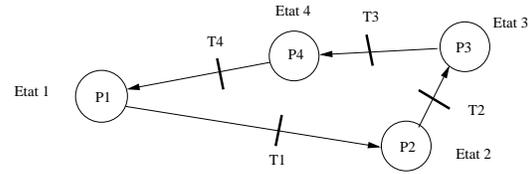


FIG. 7 – Places (états) successives et transitions (événements) obtenus à la fin du cycle 3.

le graphe, les fonctions d'appartenances et les classifieurs) diffère selon la définition de l'instant initial de la période concernée. Il en est de même selon que l'on classe ou non les individus du premier instant, lors de la première itération (dans notre cas, ce classement est effectué).

### 5.3 Classement

La Figure 8 présente les fonctions d'appartenances des marquages des différentes places  $P_n$  qui résultent de l'opération t-norme de type produit algébrique. On peut constater que ces fonctions sont non symétriques et de supports différents. Ceci montre l'adaptation de ces fonctions à l'application envisagée. L'évolution du taux de reconnaissance en classement est donnée dans la Figure 9 (avec  $X^t = X^{lr}$ ). Lorsque les degrés d'appartenances obtenus sont non nuls pour deux jetons (cf. Figure 8), les deux classifieurs correspondants peuvent être appliqués. Leurs sorties peuvent être combinées par une somme pondérée (par les degrés d'appartenances des deux jetons). Dans ce cas, l'évolution est donnée dans la Figure 10 (avec  $X^t = X^{lr}$ ). Pendant le classement, afin d'étudier la robustesse du système, un bruit peut être ajouté aux centres des classes dans toutes les positions (ce bruit choisi est Gaussien). L'évolution sans fusion mais avec bruit ( $\sigma_{centres} = 1$ ) est donnée par la Figure 11 (avec  $X^t \neq X^{lr}$ ). Le même cas avec fusion est présenté par la Figure 12. Tous ces résultats montrent que le système de reconnaissance classe correctement plus de 98 % à tout moment de la période (et même plus de 99 %, lorsqu'on autorise des combinaisons entre les résultats).

## 6 Conclusion

Dans cet article, nous avons montré comment des classes dynamiques (d'évolutions périodiques) peuvent être reconnues. Le RPF s'avère être un formalisme adéquat pour la représentation de connaissances dynamiques et en particulier pour la représentation des différents états d'un système de reconnaissance. La construction des fonctions d'appartenances des jetons et des transitions en fonction d'un ensemble d'appren-

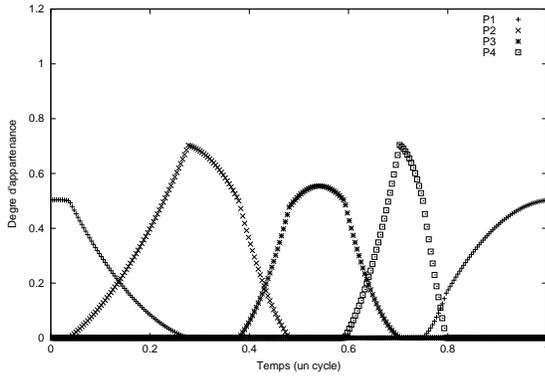


FIG. 8 – Fonctions d'appartenances des jetons  $\mu_{P_n}(t)$  obtenues pour les différentes places  $P_n$ , résultants de l'opération t-norme entre la fonction de sa transition d'entrée et celle de la place d'entrée de cette transition.

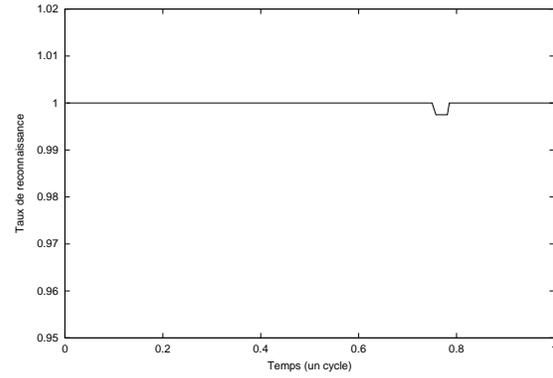


FIG. 10 – Evolution du taux de reconnaissance en classement avec fusion ( $X^t = X^{lr}$ ). Les résultats sont améliorés par la fusion, en tenant compte des deux états les plus valides.

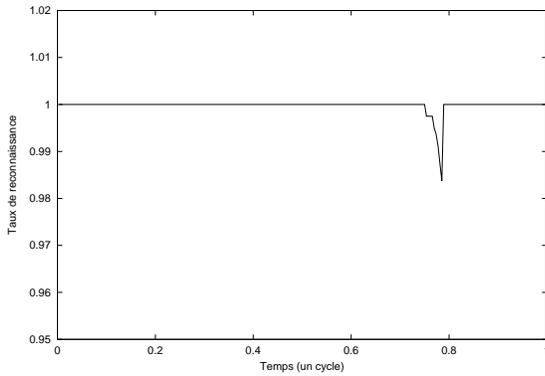


FIG. 9 – Evolution du taux de reconnaissance en classement (avec  $X^t = X^{lr}$ ). Les chutes sont minimales car le système de RdF anticipe sur l'évolution des classes.

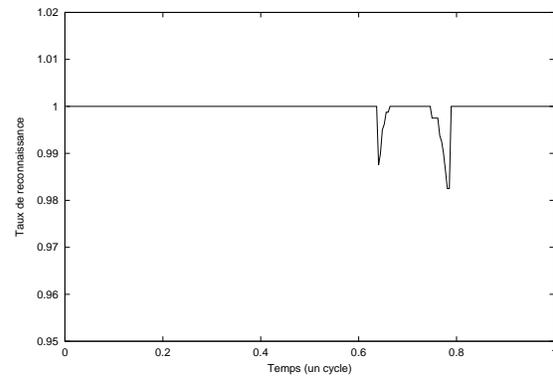


FIG. 11 – Evolution du taux de reconnaissance en classement avec bruit ( $\sigma_{centres} = 1$ ) et sans fusion ( $X^t \neq X^{lr}$ ). Peu de dégradation due au bruit : le système de RdF est robuste.

tissage permet d'anticiper sur l'évolution des classes, i.e., de prédire l'état du système de RdF adapté à l'instant considéré. De plus, la méthode est robuste au bruit. Elle permet la combinaison et la sélection de classifieurs.

Cependant, la méthode peut aussi être améliorée par une prise en compte d'informations exogènes dans les transitions et par la possibilité de gérer plusieurs futurs. Pour les problèmes apériodiques, une étude complémentaire et/ou une généralisation est nécessaire.

## Références

[1] N. Boudaoud, M. Masson, Diagnosis of transient states, a pattern recognition approach, *JESA*, Vol. 34, No. 5, pp. 689-708, 2000.  
 [2] P. Smyth, Hidden Markov Models for fault detection

in *Dynamic Systems, Pattern Recognition*, Vol. 27, No. 1, pp. 149-164, 1994.  
 [3] C. Frelicot, Un système adaptatif de diagnostic prédictif par reconnaissance des formes floues, *Thèse*, Université de Technologie de Compiègne, 1992.  
 [4] J. Cardoso, R. Valette, D. Dubois, Fuzzy Petri Nets : an overview, *13th IFAC World Congress, San Francisco, USA*, pp. 443-448, 1996.  
 [5] C.G. Looney, Fuzzy Petri nets for rule-based decision making, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 18, No. 1, pp. 178-183, 1988.  
 [6] S.M. Chen, J.S. Ke, J.F. Chang, Knowledge Representation Using Fuzzy Petri Nets, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 3, pp. 311-319, 1990.

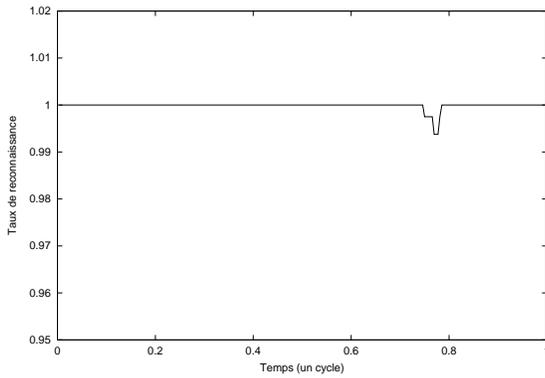


FIG. 12 – Evolution du taux de reconnaissance en classement avec bruit ( $\sigma_{centres} = 1$ ) et avec fusion ( $X^t \neq X^{lr}$ ). Lorsqu'il y a du bruit, le résultat est également amélioré.

- [7] W.T. Jong, Y.S. Shiau, Y.J. Horng, H.H. Chen, S.M. Chen, Temporal Knowledge Representation and Reasoning Techniques Using Time Petri Nets, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 29, No. 4, pp. 541-545, 1999.
- [8] R. Valette, B. Pradin-Chézalviel, F. Girault, An introduction to Petri net theory, *Studies in Fuzziness and Soft Computing*, Vol. 22, No. 1, pp. 3-24, 1999.
- [9] T. Murata, Petri nets : Properties, Analysis and Applications, *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541-580, 1989.
- [10] C.G. Looney, A.R. Alfize, Logical controls via Boolean rule matrix transformations, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 17, No. 6, pp. 1077-1082, 1987.
- [11] S. Sarkar, K.L. Boyer, Quantitative Measures of Change Based on Feature Organisation : Eigenvalues and Eigenvectors, *Computer Vision and Image Understanding*, Vol. 71, No. 1, pp. 110-136, 1998.
- [12] V. Gunes, M. Ménard, P. Loonis, A multiple classifier system using ambiguity rejection for Clustering-Classification Cooperation, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 8, No. 6, pp. 747-762, 2000.
- [13] M. Ménard, C. Demko, P. Loonis, The fuzzy c+2-means : Solving the ambiguity reject in clustering, *Pattern recognition*, Vol. 33, No. 7, pp. 1219-1237, 2000.