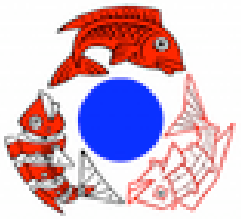


Sélection darwinienne et systèmes multi-agents

Samuel Landau
Miriad - OASIS - LIP6

Objectifs

- Conception de systèmes distribués adaptatifs
- Apprentissage distribué en ligne

Exemples d'application:

- services et réseaux informatiques
- contrôle d'un robot
- simulations distribuées (biologiques ou sociales) insuffisamment spécifiées, par exemple : évaluation de paramètres libres, évolution de comportements, émission ou validation d'hypothèses organisationnelles

Contraintes et problèmes

Contraintes

Descriptions des comportements évolués des entités :

- **Appréhendables par les concepteurs**

  par exemple assimilable à un ensemble de règles

- **Pouvant disposer d'états internes si nécessaire**

Problèmes

- **Théorique: dans tout modèle d'apprentissage distribué se pose le problème de *Credit assignment*** [Minsky 1963]
- **Pratique: comparaison des méthodes d'apprentissage utilisées**

Choix

- **Capacités adaptatives obtenue par un processus de sélection darwinienne (évolution artificielle)**
  - **Généricité, parallélisme**
  - **Peu de prérequis sur l'espace de recherche**
  - **Apporte une réponse au *Credit assignment***
- **Description des comportements par des structures intelligibles (architectures symboliques)**
- **Concevoir un cadre applicatif liant évolution artificielle et simulation multi-agent, et l'implémenter en une plate-forme**

SFERES

[Landau, Doncieux et al 2001]

- **Cadre applicatif d'évolution artificielle et de simulation multi-agent**
  - **travaux d'abstraction sur les concepts des techniques évolutionnistes (dont les algorithmes évolutionnistes)**
  - **Expérimenter plusieurs techniques évolutionnistes sur la même expérience**
  - **Réutiliser une technique évolutionniste dans plusieurs expériences**
- **Plate-forme implantée en C++ sur Linux et sur les BSD libres**
  - **Echéancier discret, interface avec un moteur de simulation 3D physique-réaliste, gestion fine des statistiques…**
  - **Environ 80 classes abstraites/80 000 lignes de code, et 400 classes dérivées/400 000 ldc**
  - **Les principaux algorithmes évolutionnistes et architectures évoluables d'agent sont implantées**

Sélection darwinienne

Principes [Darwin 1859]

- **Individus en compétition pour des ressources limitées**
  - **Sélection**
- **Ressemblance entre parents et enfants (hérédité)**
  - **Variation**
- **Hypothèse d'indépendance entre les causes de variations et celles de sélections**
  - **Variations « aléatoires »**
  - **Modèle « hasard - sélection »**
- **Gradualisme**
- **Darwinisme synthétique** [Huxley/Wright 1942]
  - **Introduction de la génétique**

Algorithmes évolutionnistes

- **Méthodes d'optimisation inspirées du modèle darwinien synthétique**
- **Sélection suivant la valeur d'une fonction d'évaluation (*fitness*)**

\+ **Le génome peut coder pour des structures**

\+ **Peu de prérequis sur l'espace de recherche**

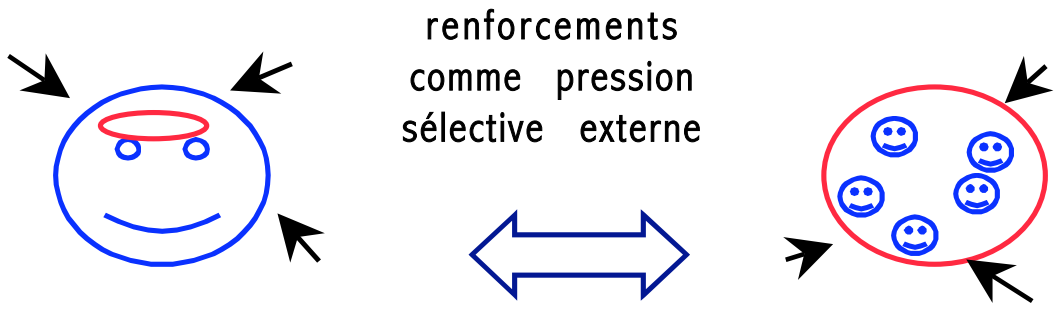
\- **Évolution hors ligne**

\- **Pas de solution générique pour le *credit assignment***

\- **Coût élevé en nombre de présentation d'exemples (≈temps de calcul)**

Dualité organisme / écosystème

- **Population en croissance et compétition sur les ressources**
  - ➢ **Processus de sélection darwinienne**
- **Les renforcements distribués au système influent sur la disponibilité des ressources**
  - ➢ **Pression sélective induite**



Métaphore biologique de « l'ontophylogenèse » [Kupiec 1981]

Modèles ontophylogénétiques

Comparaison avec les algorithmes évolutionnistes

Algorithmes Évolutionnistes

1. Centralisé
2. Évaluation « statique »
3. Phylogenèse
4. Fitness explicite
5. Hors ligne
6. Cycle exploration/exploitation

Modèles ontophylogénétiques

1. Distribué
2. Évaluation « dynamique »
3. Ontogenèse

   \+ phylogenèse (échelle inf.)
4. Fitness implicite
5. En ligne
6. Exploration ***et*** exploitation

Modèles ontophylogénétiques
Cas particuliers

Approche « Parisienne »

Systèmes de classeur (LCS) « Michigan style »

- **Les individus sont les classeurs du système**
- **Nombre de classeurs et d'états internes (le cas echéant) fixé à l'avance**

Systèmes immunitaires artificiels (AIS)

- **Les individus sont des vecteurs (« anticorps »)**
- **Les ressources *constituent* les renforcements**

Modèles ontophylogénétiques
Propriétés

- **Moins de présentations d'exemples nécessaires**
- **Possibilité de distribuer les critères d'évaluation sur la population**
- ***Credit assignment* implicite (*via* la sélection darwinienne)**
  - **Extension et unification de l'apprentissage par renforcement et de l'évolution artificielle pour un système distribué**

Avec des structures :

- **Plus souples que les LCS et AIS à l'échelle individuelle (par ex. nombre de règles / d'états internes non fixé a priori)**
- **Plus souples que les LCS et AIS à l'échelle collective**

  (par ex. populations non-homogènes / interactions plus riches)

Propriétés recherchées

- **Puissance d'expression**
- **« Quasi continuité » du code génétique**
  - **cadre gradualiste de la sélection darwinienne**
  - **de petites variations du génotype doivent induire la plupart du temps des petites variations de la structure**
    - **Neutralité** [Kimura 1983]
    - **Redondance**
    - **Épistasie faible**
- **Pouvoir générer des structures appréhendables**

  pour faciliter l'analyse, les manipulations et l'amorçage

Les codes génétiques existants ne vérifient pas toutes ces propriétés simultanément

Codes génétiques
directs

Le génotype *est* la structure

Principales instances:

- **Programmation génétique** [Koza 1992]

  arbres

- **Programmation évolutionniste** [Fogel et al 1966]

  machines à états finis, réseaux de neurones, graphes,…

+ **Puissance d'expression**

+ **Structures appréhendables**

– **Pas de redondance**

– **Épistasie forte**

– **Neutralité faible**

  ➢ **Nécessité de biaiser fortement les opérateurs génétiques**

Codes génétiques
indirects

Construction **de la structure à partir du « programme » génétique**

➢ **langages de construction, basés pour la plupart sur :**

- **une grammaire non contextuelle**
- **ou des règles de réécriture de type $\mathcal{L}$-systems**

réseaux de neurones, circuits électroniques

\+ **Puissance d'expression**

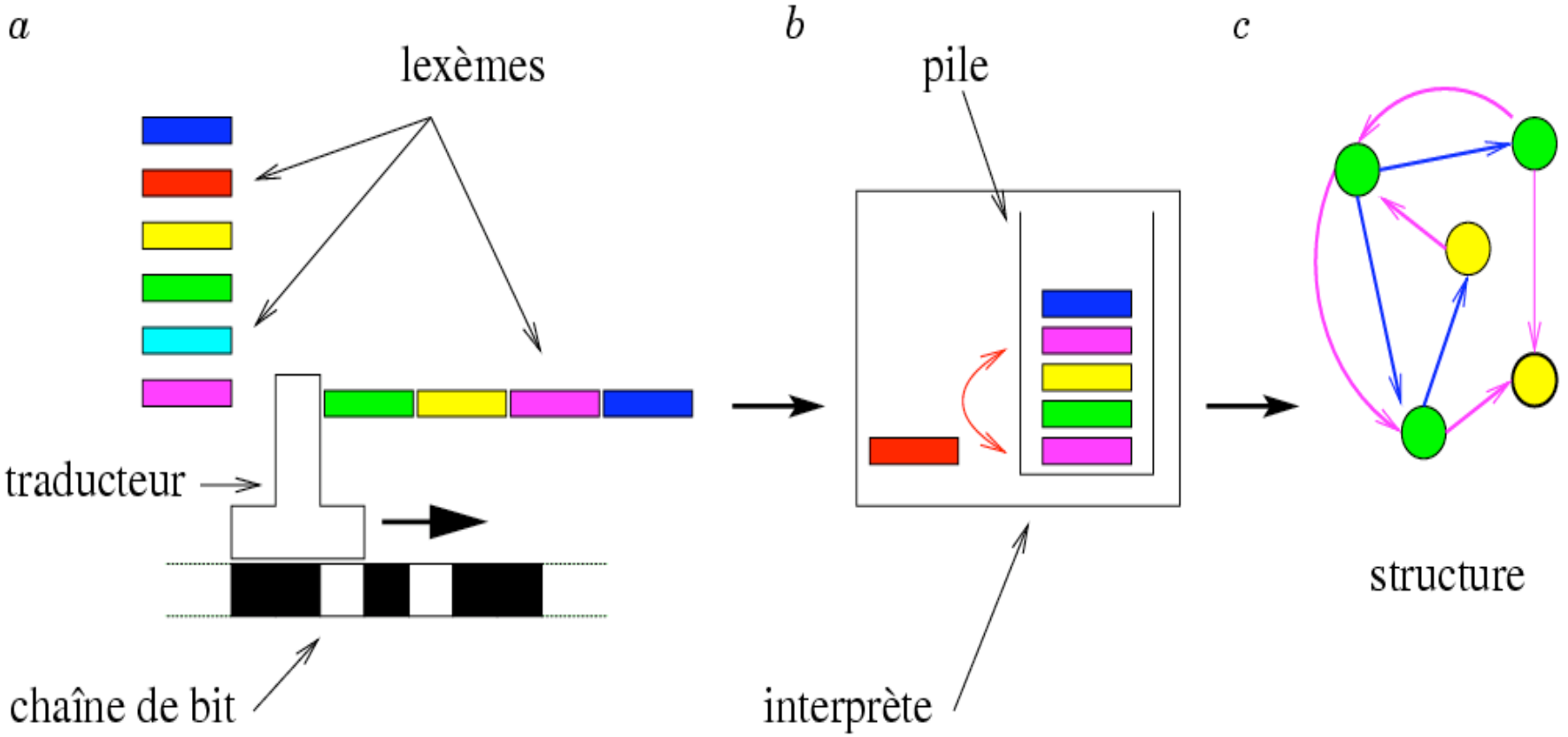
\+ **Structures appréhendables**

o **Redondance et neutralité suivant grammaire**

– **Épistasie hiérarchique liée à l'application des règles de grammaire**

➢ **Nécessité pour les opérateurs génétiques de respecter les règles grammaticales**

Code indirect basé sur un langage à pile

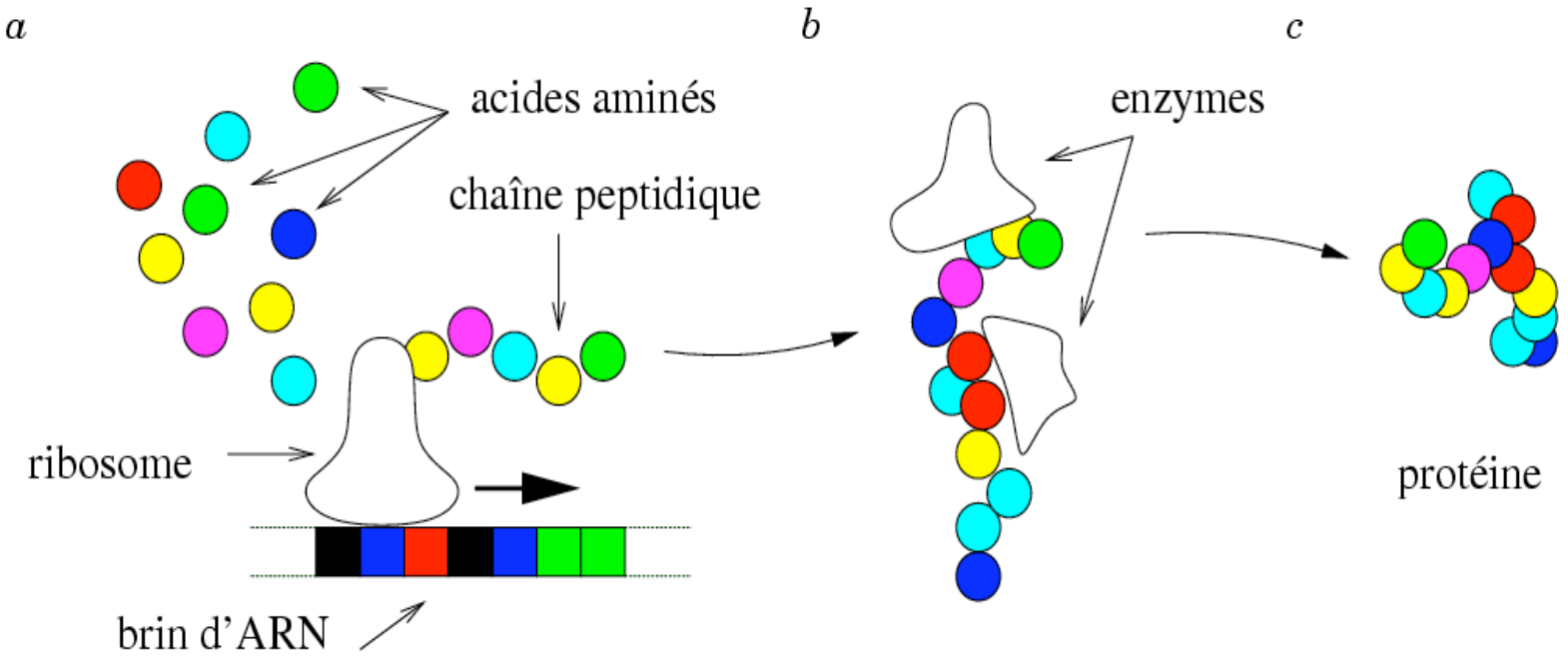


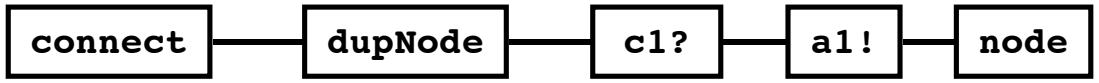
Stack-Based Gene Expression
Propriétés

- **Opérateurs classiques des AG, et même plus (insertion/délétion) car**
- **Toute chaîne est interprétable**
  - ➢ **Pas de biais *nécessaire* sur les opérateurs génétiques pour respecter une syntaxe**
- **Neutralité et redondance**
  - **du code génétique (surjection)**
  - **à l'interprétation**

    par ex.: l'ordre peut ne pas importer pour certaines instructions
  - **plusieurs façon de coder les mêmes sous-structures**
  - **structures différentes mais mêmes sémantique (dépend de l'environnement)**
- **Épistasie**
  - **Localité des instructions de la pile**
- **Grammaire *contextuelle*, « distribuée » sur les lexèmes**

Métaphore de la synthèse de protéines

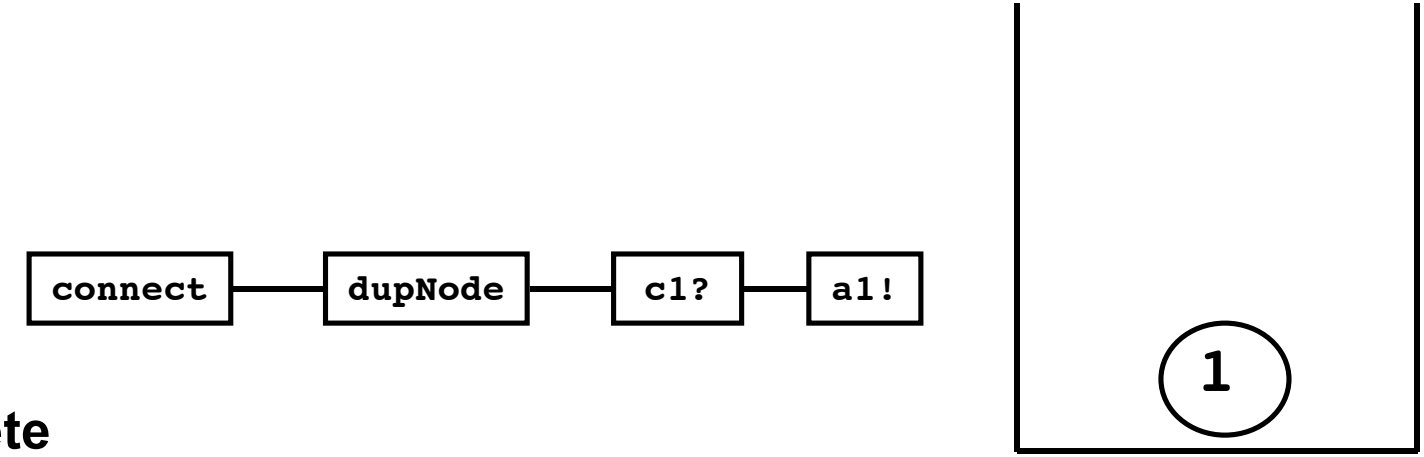




interprète

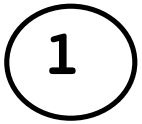
ATN

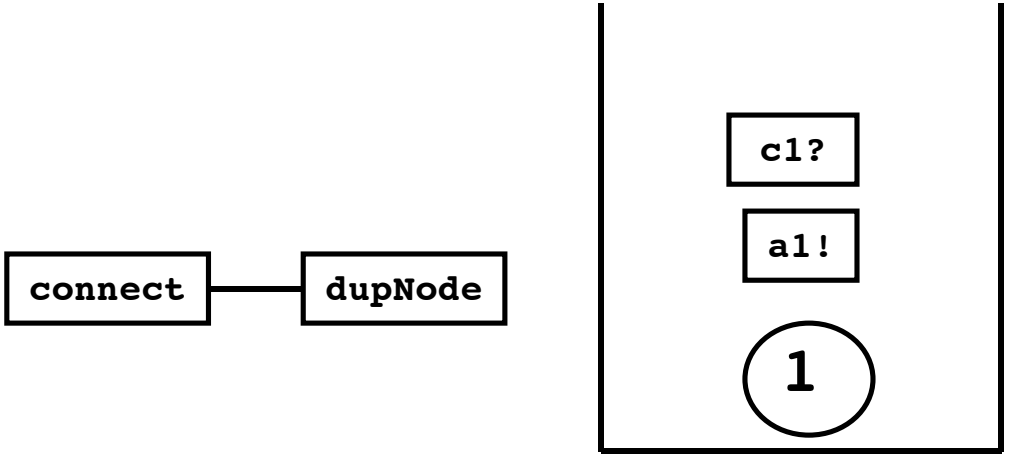
`<vide>`



interprète

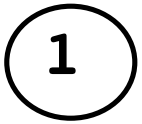
ATN

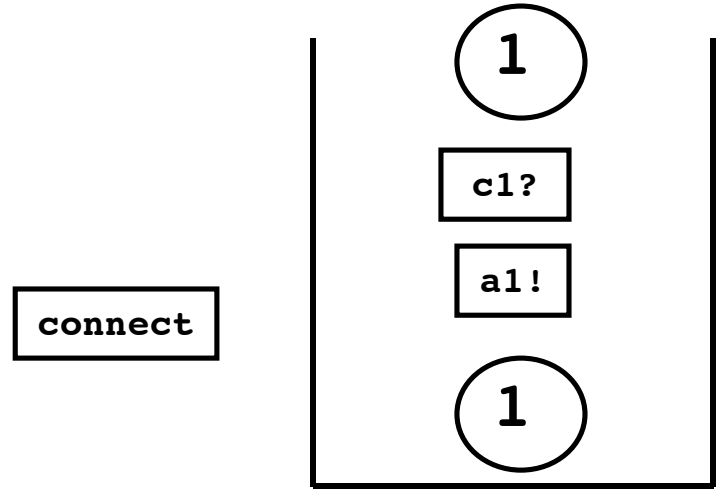




interprète

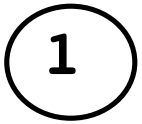
ATN

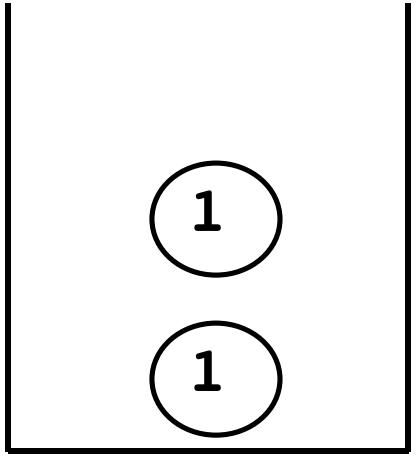




interprète

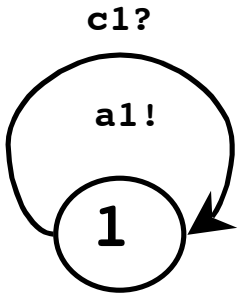
ATN





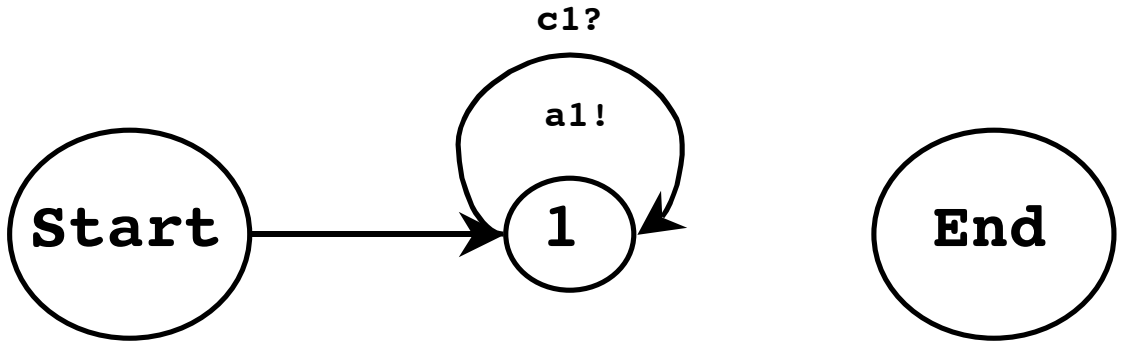
interprète

ATN

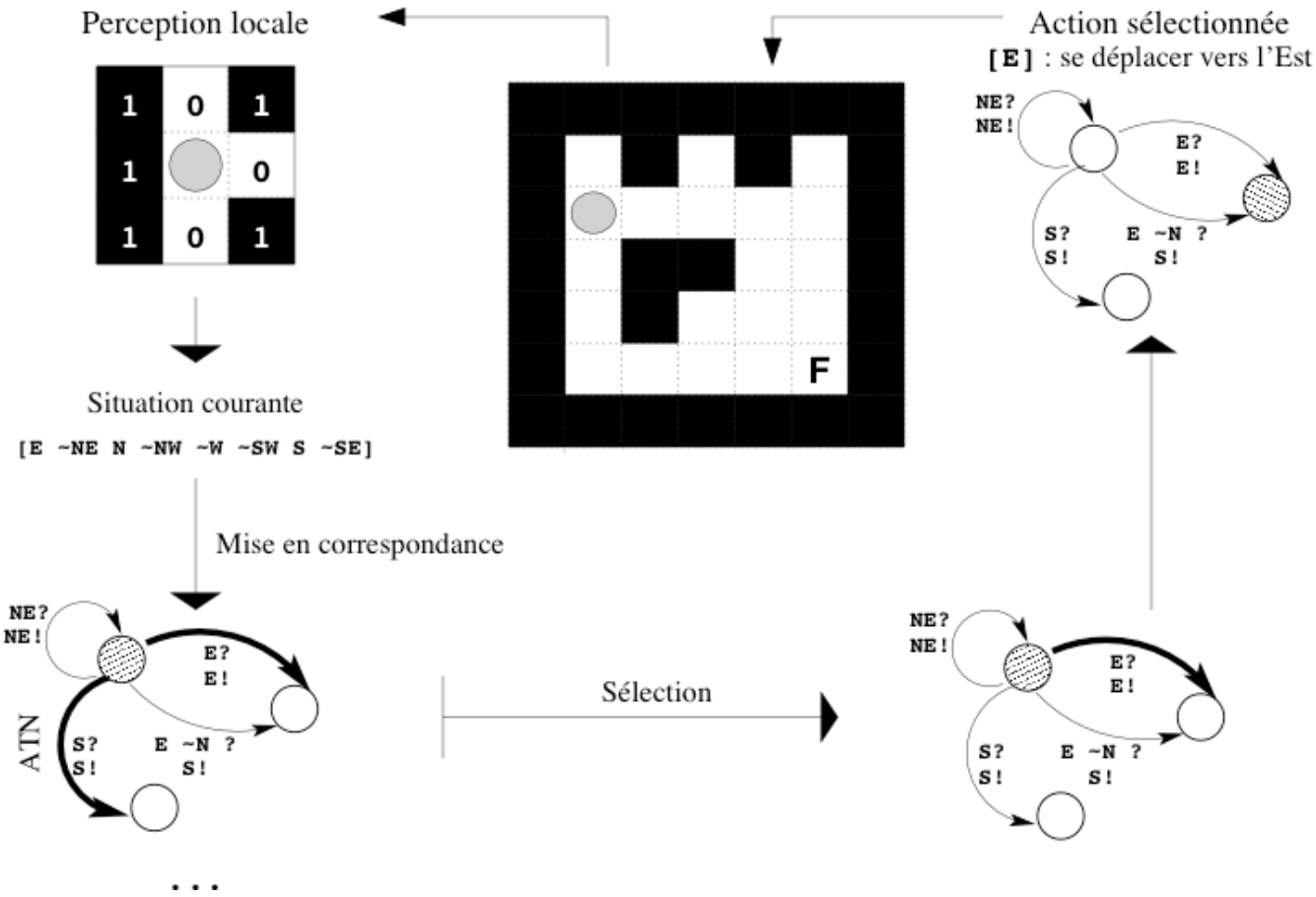


interprète

ATN



Expérimentations individuelles
Fonctionnement

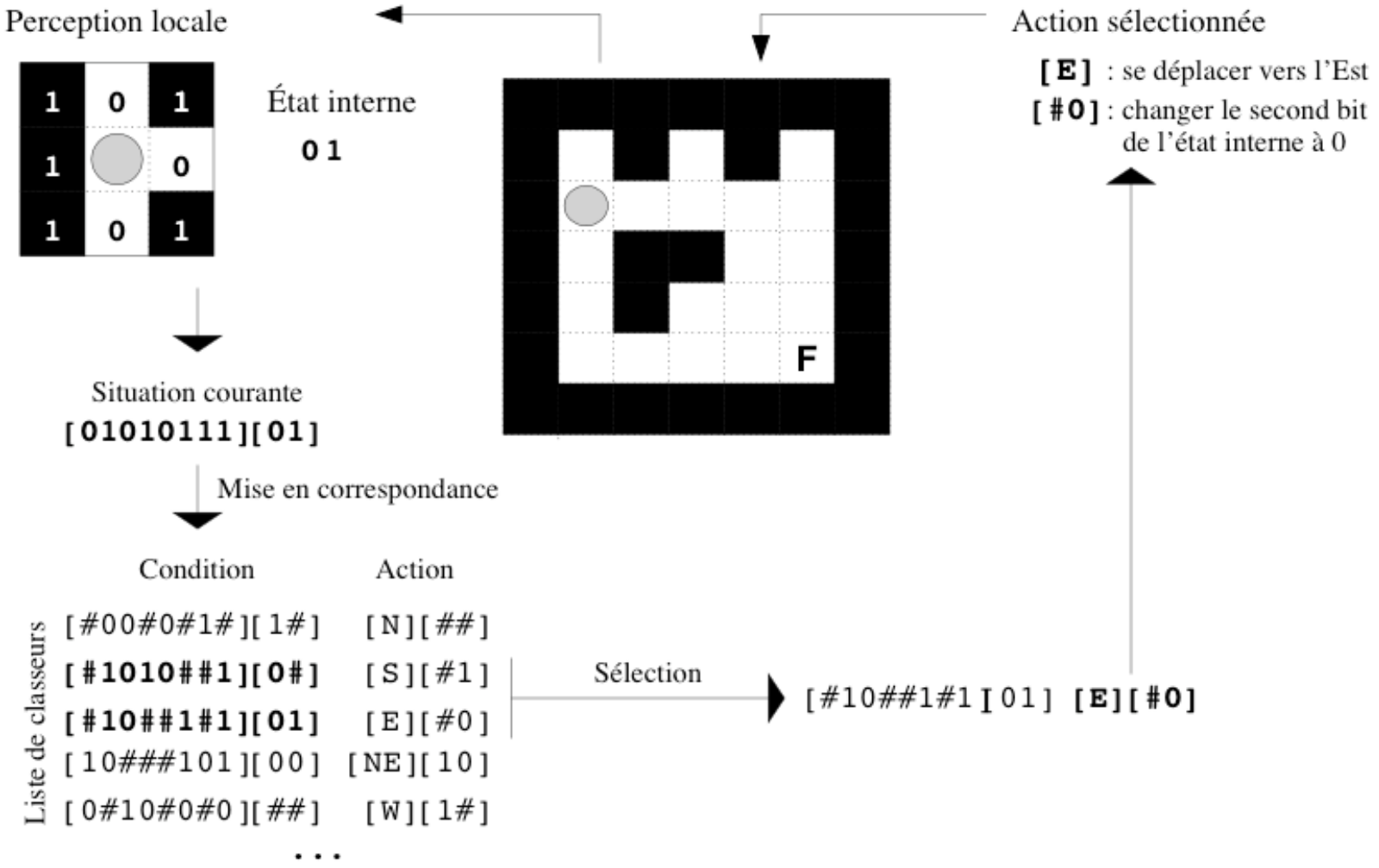


- **Conditions: N?/~N?, NE?/~NE?, …**
- **Actions : N!, NE!,…**

Comparaison avec les LCS
sur des problèmes non-markoviens

- **Dans les modèles distribués, la localité des perceptions et actions des agents induisent souvent de devoir résoudre individuellement des POMDP**
- **ATNoSFERES *peut* résoudre individuellement des problèmes markoviens (expériences non décrites), mais de façon beaucoup moins efficace que les LCS (du fait de l'algorithme génétique)**
- **Si on ne sait pas à l'avance si le problème individuel sera markovien ou non, ATNoSFERES présente *a priori* l'avantage de ne pas nécessiter plus d'hypothèses**
  - **Comparaisons sur des bancs d'essais non-markoviens avec les principaux LCS permettant d'attaquer des problèmes non-markoviens : XCSM (LCS à registres) et ACS (LCS pouvant exécuter des « chaînes comportementales »)**

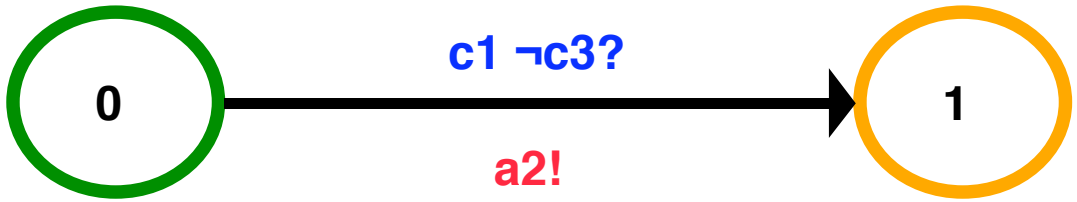
Comparaison avec les LCS
Fonctionnement de XCSM



Comparaison avec les LCS
Équivalence formelle avec XCSM

- Étiquettes de condition / action sur les arcs (en limitant le nombre d'actions) ≈ conditions / actions externes dans les règles
- Noeud ≈ état interne
- Origine d'un arc ≈ condition interne
- Extrémité d'un arc ≈ action interne

ATN

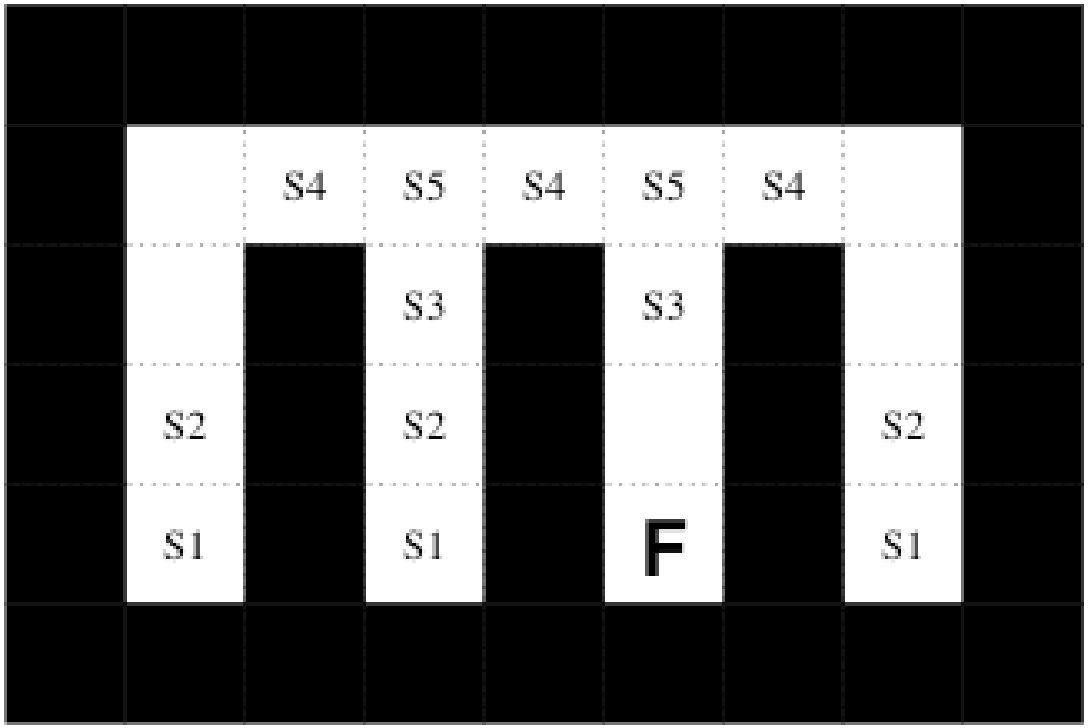


XCSM

conditions				actions			
externe			interne	externe		interne
c1	c2	c3	s1	a1	a2	g1
1	#	0	0	0	1	1

Environnement Maze10

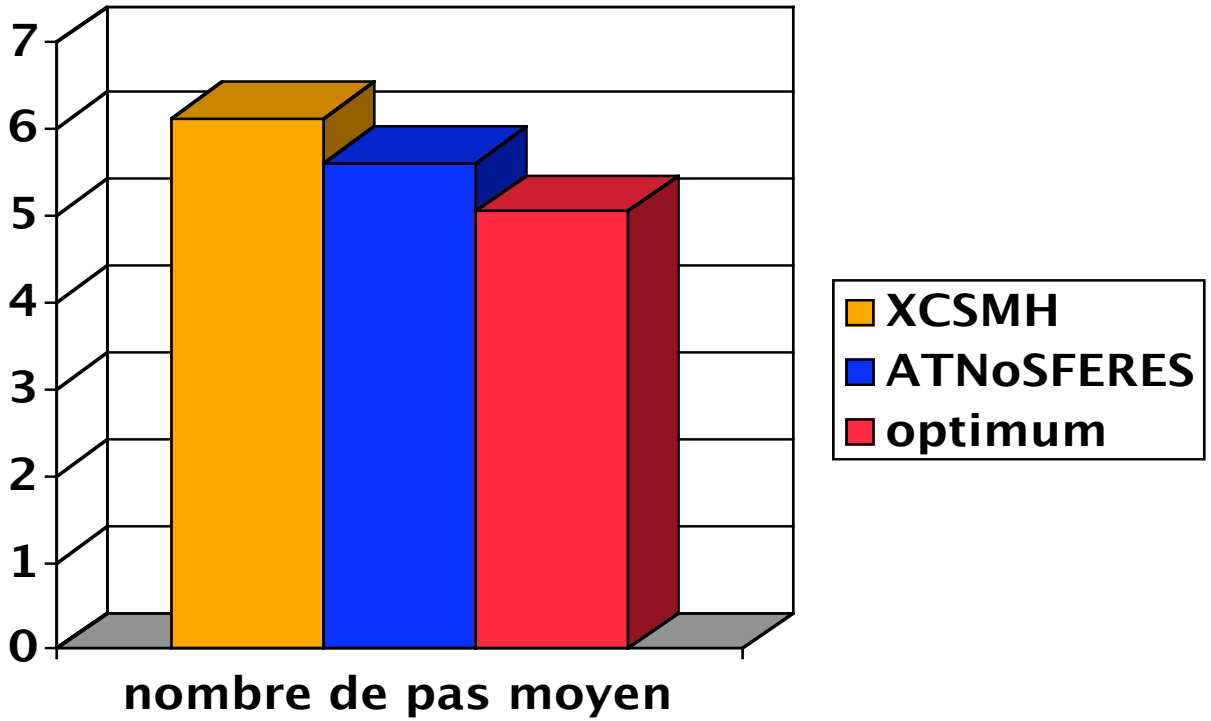
[Lanzi 1998]



environnement non-markovien :

- **13 cases présentant des alias perceptuels**
- **5 cases non ambiguës ; 1 case « food » (sortie), perçue**

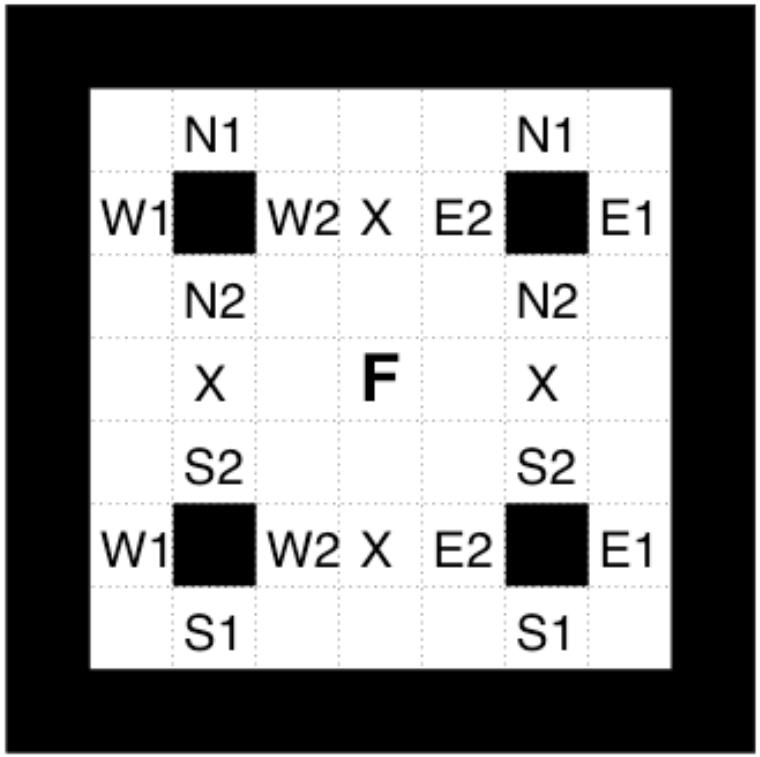
Comparaison
résultats Maze10



- **XCSMH : version de XCSM « hiérarchique » optimisée pour gérer plus efficacement les états internes**
- **Les résultats pour XCSM (15,1) n'ont pas été portés pour ne pas « écraser » les histogrammes**

Environnement E1

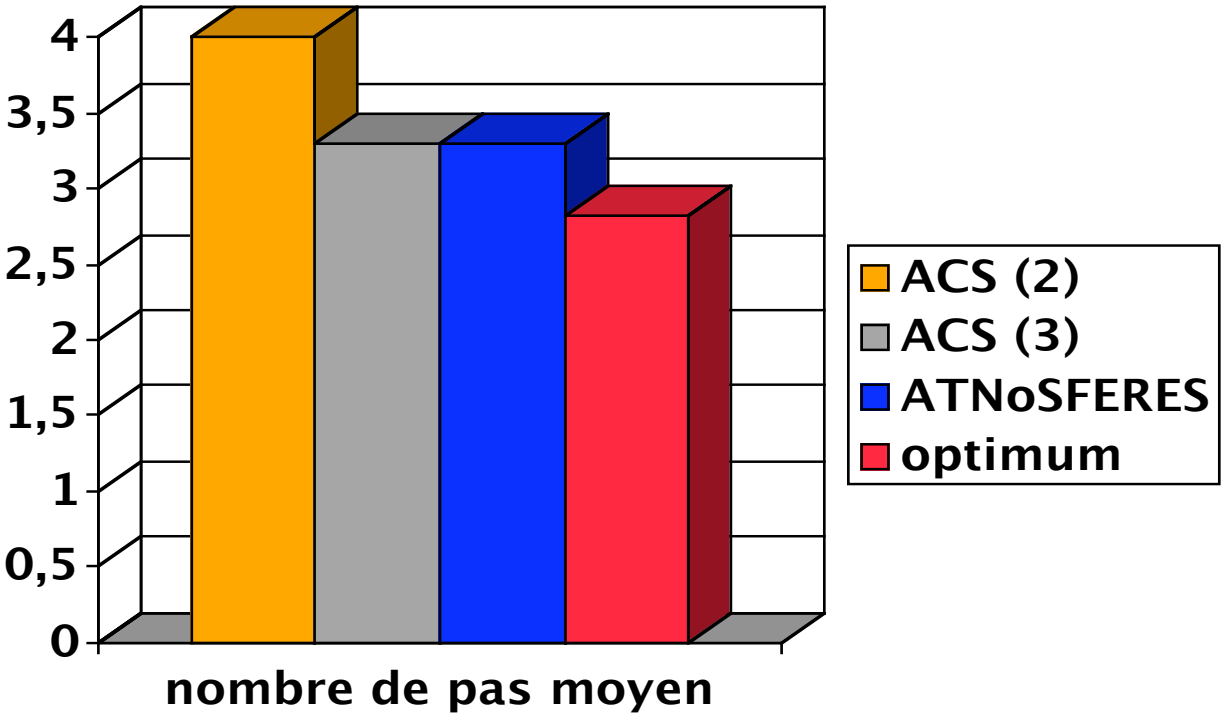
[Métivier et Lattaud 2002]



environnement non-markovien :

- **20 cases présentant des alias perceptuels**
- **24 cases non ambiguës ; 1 case « food » (sortie), perçue**

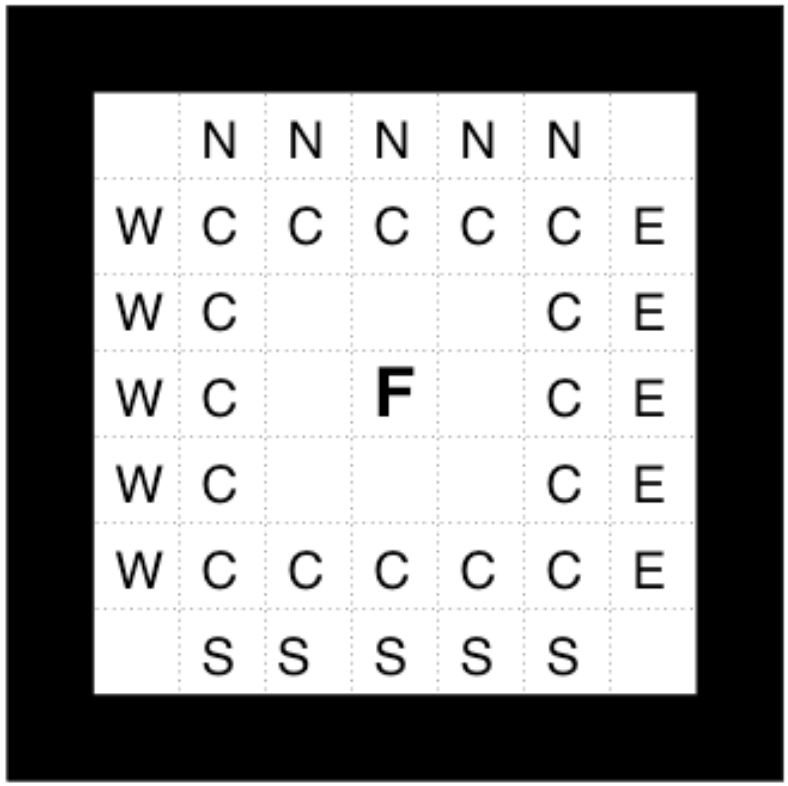
Comparaison
résultats E1



- **ACS nécessite de fixer au préalable le longueur maximale des « chaînes comportementales » (paramètre BS max), qui influe beaucoup sur la vitesse d'apprentissage du modèle du monde**
- **Plus d'améliorations au-delà de BS max=3**

Environnement E2

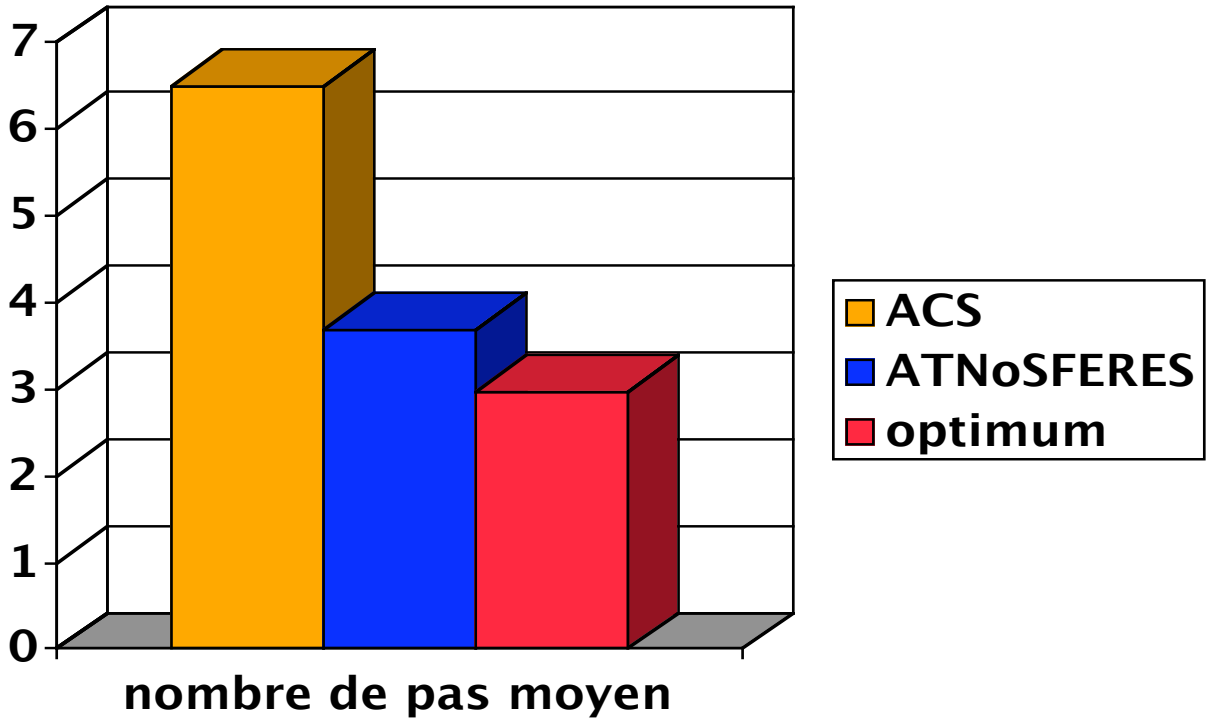
[Métivier et Lattaud 2002]



environnement non-markovien :

- **36 cases présentant des alias perceptuels**
- **22 cases non ambiguës ; 1 case « food » (sortie), perçue**

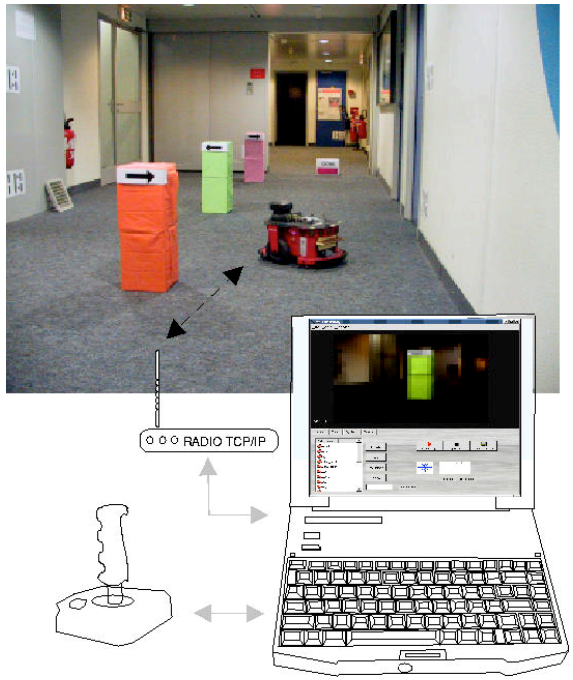
Comparaison
résultats E2



- **Métivier et Lattaud rapportent que faire varier BS max ne change pas grand-chose aux performances du système**

Première expérimentation collective

Enseigner un comportement à un robot pas démonstration



traitement

démonstrations

réalisation

Modèle naïf : un SMA contrôle le robot. Chaque agent perçoit la couleur d'un pixel de l'image captée par la caméra du robot, et fait une proposition d'action. Les actions sont agrégées par un système de vote, et l'action résultante est exécutée par le robot. L'utilisateur distribue le renforcement au système soit explicitement, soit implicitement en pilotant le robot.

Première expérimentation collective

Résultat : échec

Problème *trop* complexe pour des études préliminaires de système ontophylogenétiques : modèle asynchrone pour ce qui est du développement / la robotique en environnement réel implique des perceptions bruitées et le traitement des perceptions visuelles est un problème vaste en soi…

A cependant permis de soulever des questions :

- **celle du synchronisme des agents, qui ne se pose pas pour les LCS/AIS**
- **celle de la régulation du développement et de la pression sélective**

Les prochaines expériences sont en simulation, en substituant à ATNoSFERES et à l'algorithme génétique un modèle distribué sur des automates, apprenant par renforcement

Conclusion

Résultats

- SFERES validé (plusieurs techniques/expériences implantées)
- SBGE et ATNoSFERES validés dans le cadre individuel
- Observations expérimentales : parcimonie, régulations implicites

Travaux actuels

- Étude du modèle de contrôle distribué en simulation

Perspectives

- Apprentissage multicritère, distribué sur une population d'automates possiblement non-homogène
- Auto-observation et capacité du système à agir sur sa propre activité, afin de créer des bassins d'attraction le long du front de Pareto
- Application dans des modèles sociaux et naturels (CAS)
- Retour à la robotique quand le modèle sera dominé en simulation