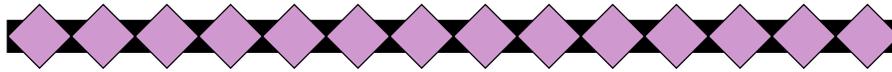


IFT-17586

Intelligence artificielle I



Apprentissage

Benoît Potvin

Département d'informatique et de génie logiciel
Faculté des sciences et de génie, Université Laval
Été 2003

Plan



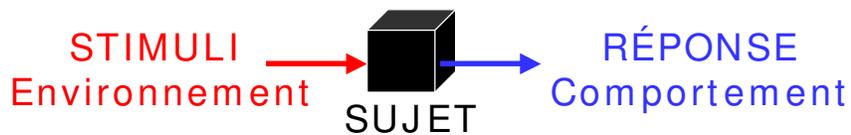
- ◆ Apprentissage humain
 - Approche behavioriste
 - Approche cognitiviste
- ◆ Apprentissage automatique
 - Approche symbolique
 - Approche connexionniste
 - Approche sociale et émergente

Apprentissage humain

- ◆ Approche behavioriste (< ~1950)
 - Pavlov, Watson, Guthrie, Thorndike, Hull, Skinner
- ◆ Approche cognitive (> ~1950)
 - Hebb, Tolman, Bruner, Piaget

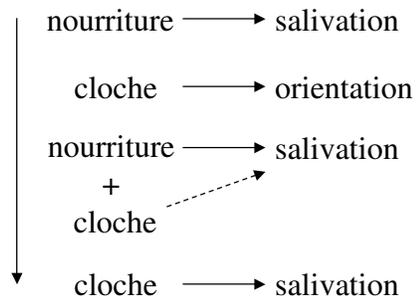
Apprentissage humain : approche behavioriste

- ◆ On s'intéresse aux faits observables et non aux spéculations
 - Le comportement est observable
 - Les processus mentaux sont spéculatifs
- ◆ Processus internes : boîte noire



Apprentissage humain : approche behavioriste

◆ Conditionnement classique (Pavlov)



© pigeon.psy.tufts.edu

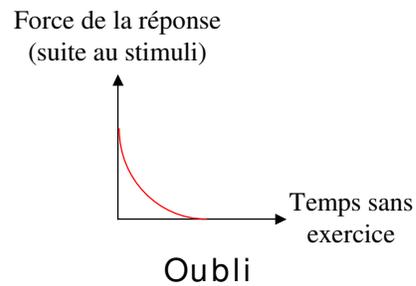
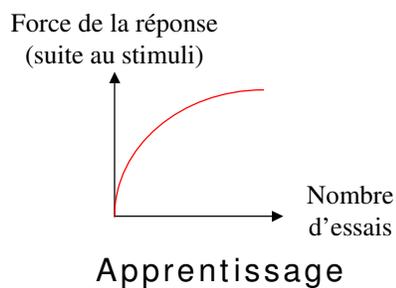
STIMULI **RÉPONSE**

© Capus, Potvin et Tourigny, 2003

5

Apprentissage humain : approche behavioriste

◆ Apprentissage



© Capus, Potvin et Tourigny, 2003

6

Apprentissage humain : approche behavioriste

◆ Loi de l'Effet (Thorndike, 1913)

- L'apprentissage d'une connexion S-R dépend du nombre d'essais mais surtout des conséquences de la réponse

➔ La pratique favorise l'apprentissage
(essais-erreurs)

➔ La récompense favorise l'apprentissage
(renforcement)

Apprentissage humain : approche behavioriste

◆ Conditionnement opérant (Skinner)

- La réponse n'est pas « élicitée » comme dans le conditionnement classique
- Les conséquences du comportement déterminent la probabilité que le comportement se produise de nouveau

Apprentissage humain : approche behavioriste

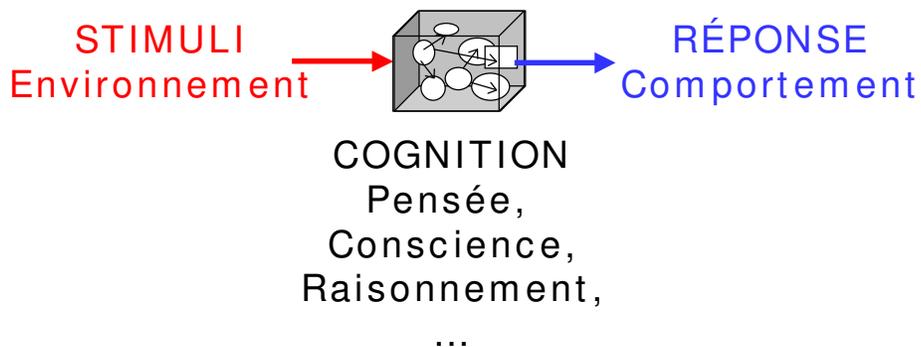
◆ Conditionnement opérant (Skinner)

– Apprentissage

- Par renforcement
 - Comportement + événement plaisant
 - Comportement – événement non-plaisant
- Par punition
 - Comportement - événement plaisant
 - Comportement + événement non-plaisant

Apprentissage humain : approche cognitiviste

◆ Cognitivism



Apprentissage humain : approche cognitive

- ◆ Une transition (Tolman)
 - Les rats ont un « plan cognitif »

(c) Lefrançois, 1995

© Capus, Potvin et Tourigny, 2003

11

Apprentissage humain : approche cognitive

- ◆ Processus mentaux de haut-niveau (Hebb)
 - Base neurobiologique
 - Neurones

(c) Lefrançois, 1995

© Capus, Potvin et Tourigny, 2003

12

Apprentissage humain : approche cognitive

- ◆ Processus mentaux de haut-niveau (Hebb)
 - Assemblée de neurones, séquence de phase

(c) Lefrançois, 1995

© Capus, Potvin et Tourigny, 2003

13

Apprentissage humain : approche cognitive

- ◆ Processus mentaux de haut-niveau (Hebb)
 - Pensée
 - Perception → Activation → comportement
 - Apprentissage
 - Activation simultanée → Tendance à association
 - Attention
 - sélectivité dans les perceptions

© Capus, Potvin et Tourigny, 2003

14

Apprentissage humain : approche cognitive

- ◆ Évolution de la représentation (Bruner)
 - R. Enactive (motrice)
 - R. Iconique (sensitive)
 - R. Symbolique (intellectuelle)

Apprentissage humain : approche cognitive

- ◆ Théorie de la catégorisation (Bruner) : Catégorie =
 - Un concept
 - Lorsque 2 objets distincts sont traités de façon équivalente
 - Une séquence de phase (Hebb)
 - Une règle pour classifier un objet
 - Attributs, organisation, importance, limites

Apprentissage humain : approche cognitive

◆ Théorie de la catégorisation (Bruner)

- Apprendre un concept
 - Découvrir l'existence d'une « classe »
 - ex : il existe des champignons comestibles et des champignons non-comestibles
- Atteindre un concept
 - Découvrir les attributs qui distinguent les membres d'une classe

© Capus, Potvin et Tourigny, 2003

17

Apprentissage humain : approche cognitive

◆ Théorie de la catégorisation (Bruner)

- Les frontières entre catégories sont floues
 - ex : petit, grand
- Les concepts dans la mémoire
 - prototype (ex : véhicule)
 - ensemble d'exemples (ex : arbre)

© Capus, Potvin et Tourigny, 2003

18

Plan

- ◆ Apprentissage humain
 - Approche behavioriste
 - Approche cognitiviste
- ◆ Apprentissage automatique
 - Approche symbolique
 - Approche connexionniste
 - Approche sociale et émergente

Apprentissage automatique

- ◆ Importance de l'apprentissage
 - Évolution de la base des connaissances
 - Agents doivent apprendre de leur expérience
 - Solution au processus de l'acquisition des connaissances :
 - quantité minimum de connaissances
 - apprentissage à partir d'exemples du domaine

Apprentissage automatique

◆ Importance de l'apprentissage

- Généralisation de l'expérience : plus que la répétition d'une même tâche mais la répétition de tâches similaires
- But général : apprendre en recherchant dans un espace de concepts possibles afin de trouver une généralisation acceptable

Apprentissage automatique

◆ 3 approches principales :

- SYMBOLIQUE : algorithmes d'induction
- CONNEXIONNISTE : réseaux de neurones
- SOCIALE ET ÉMERGENTE : algorithmes génétiques

1) Approche symbolique

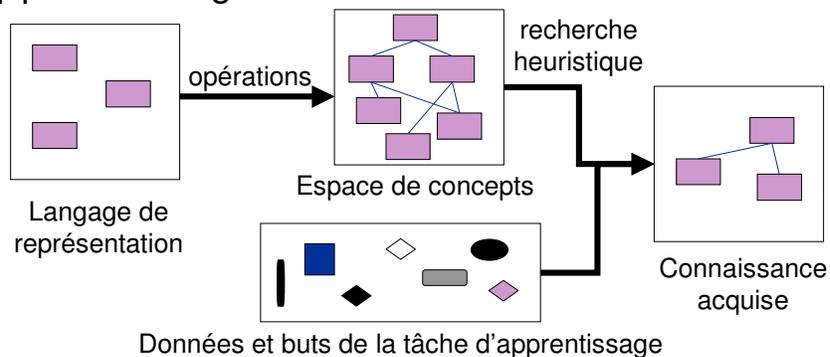
- ◆ Processus de généralisation à partir d'un ensemble réduit d'exemples
- ◆ Connaissances insuffisantes pour garantir une généralisation optimale ⇒ problème de l'induction
- ◆ Généralisation basée sur des heuristiques
- ◆ Utilisation de la base du système qui contient les connaissances du domaine représentées explicitement

© Capus, Potvin et Tourigny, 2003

23

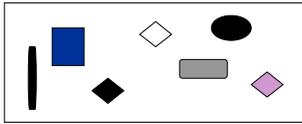
Approche symbolique :

- ◆ Modèle général du processus d'apprentissage



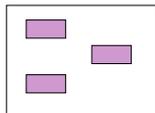
© Capus, Potvin et Tourigny, 2003

24



Données et buts de la tâche d'apprentissage

- ◆ Un ensemble d'exemples positifs et/ou négatifs d'une classe source
⇒ une définition générale pour reconnaître les futures instances de la classe
- ◆ Un seul exemple et une base de connaissances spécifique du domaine
⇒ un concept général
- ◆ Un ensemble d'exemples non classifiés
⇒ des catégorisations utiles



Langage de représentation

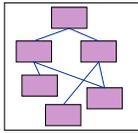
- ◆ Utilisation d'un mode de représentation des connaissances
- ◆ Exemple en logique des prédicats :
 - deux instances de balles :
 $\text{taille}(o1, \text{petite}) \wedge \text{couleur}(o1, \text{rouge}) \wedge \text{forme}(o1, \text{ronde})$
 $\text{taille}(o2, \text{grande}) \wedge \text{couleur}(o1, \text{rouge}) \wedge \text{forme}(o1, \text{ronde})$
 - définition générale du concept « balle » :
 $\text{taille}(O, T) \wedge \text{couleur}(O, \text{rouge}) \wedge \text{forme}(O, \text{ronde})$

Ensemble d'opérations

- ◆ Manipulation des connaissances :
 - généralisation ou spécialisation d'expressions symboliques
 - ajustement de poids dans un réseau de neurones
 - modification des représentations utilisées

Ensemble d'opérations

- ◆ Exemple de généralisation :
 $\text{taille}(o1, \text{petite}) \wedge \text{couleur}(o1, \text{rouge}) \wedge \text{forme}(o1, \text{ronde})$
 $\Rightarrow \text{taille}(o1, X) \wedge \text{couleur}(o1, \text{rouge}) \wedge \text{forme}(o1, \text{ronde})$
 $\text{taille}(o1, \text{petite}) \wedge \text{couleur}(o1, X) \wedge \text{forme}(o1, \text{ronde})$
 $\text{taille}(o1, \text{petite}) \wedge \text{couleur}(o1, \text{rouge}) \wedge \text{forme}(o1, X)$
 $\text{taille}(X, \text{petite}) \wedge \text{couleur}(X, \text{rouge}) \wedge \text{forme}(X, \text{ronde})$



Espace de concepts

- ◆ Mode de représentation des connaissances et opérations
⇒ espace de concepts
- ◆ Complexité de l'espace : mesurer la difficulté du problème d'apprentissage

Recherche heuristique

- ◆ Choisir le premier exemple comme concept candidat
- ◆ Généraliser cet exemple
- ◆ Inclure les exemples suivants

Algorithme ID3

Fonction arbre_inductif(Exemples, Propriétés)

Début

Si toutes les entrées de Exemples sont de la même classe

Alors retourner une feuille étiquetée à cette classe

Sinon Si Propriétés est vide

Alors Retourner une feuille étiquetée avec la disjonction de toutes les classes dans Exemples

Sinon Début

Sélectionner une propriété P

Racine de l'arbre courant ← P

Effacer P de Propriétés

Pour chaque valeur V de P

Début

Créer une branche étiquetée à V

Partition_V = éléments de Exemples avec la valeur V pour P
arbre_inductif(Partition_V, Propriétés) relié à la branche V

Fin

Fin

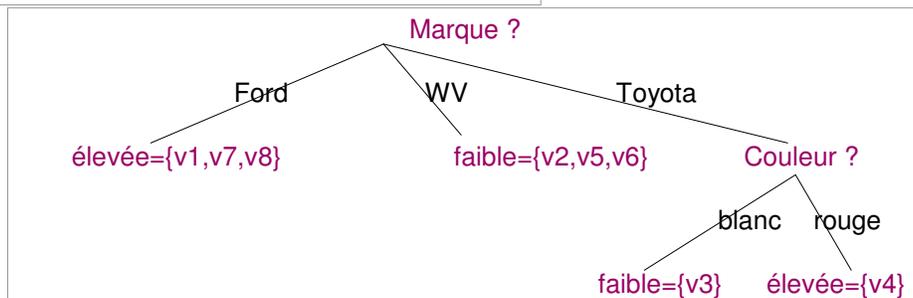
Fin

Algorithme ID3 : exemple

	Marque	Couleur	Puissance
v1	Ford	rouge	élevée
v2	WV	noir	faible
v3	Toyota	blanc	faible
v4	Toyota	rouge	élevée
v5	WV	noir	faible
v6	WV	rouge	faible
v7	Ford	blanc	élevée
v8	Ford	blanc	élevée

- Échantillon d'éléments correctement classifiés en fonction de catégories sources

- Construction d'un arbre de décision



Approche symbolique

◆ Arbre de décision

- Disjonction de règles pour classifier un objet
- Définition d'un concept
- Avec ID3, l'arbre produit est le plus petit pouvant classifier correctement tous les exemples d'entraînement

2) Approche connexionniste

- ◆ Pas de symboles
- ◆ Réseaux neuronaux = systèmes de neurones artificiels interconnectés
- ◆ Connaissances implicites dans l'organisation et l'interaction de ces neurones
- ◆ N'ajoute pas de connaissances à la base mais modifie la structure d'ensemble

Composants d'un neurone artificiel

◆ Un neurone biologique

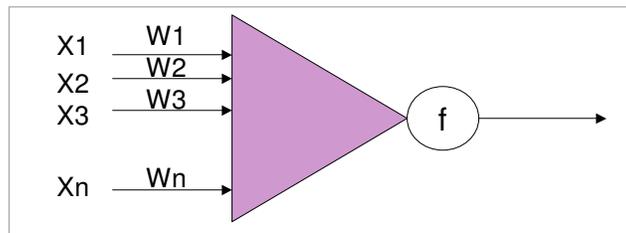
- Dendrites
- Axone
- Noyau

Composants d'un neurone artificiel

◆ Un neurone informatique

- Dendrites = entrées du neurone
- Axone = sortie du neurone
- Noyau = sommation et fonction d'activation

Composants d'un neurone artificiel



- X_i = Signaux d'entrée : données de l'environnement ou de l'activation d'un autre neurone
- W_i = Ensemble de poids : force de la connexion
- $\sum W_i X_i$ = Niveau d'activation (somme des poids d'entrée)
- f = Fonction seuil (calculer la sortie du neurone : au-dessus ou au-dessous du seuil)

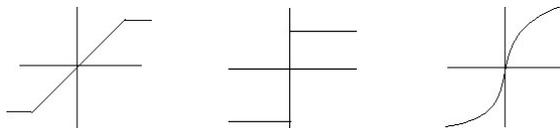
© Capus, Potvin et Tourigny, 2003

37

Composants d'un neurone artificiel

◆ Type de fonction d'activation

- Linéaire
- Seuil
- Sigmoide



© Capus, Potvin et Tourigny, 2003

38

Exemple

◆ ET logique

- Le neurone a 2 entrées qui peuvent prendre la valeur 0 ou 1
- Le poids multiplicatif de chacune de ces entrées est 0.5
- La fonction d'activation se déclenche lorsque la valeur de la sommation est plus grande que 0.5

E1	E2	Valeur somme	Sortie
0	0	0	0
0	1	0.5	0
1	0	0.5	0
1	1	1	1

Réseaux de neurones

- ◆ Systèmes de neurones artificiels
- ◆ Une topologie du réseau : patron de connexions entre les neurones individuels
- ◆ Algorithme d'apprentissage : perceptron, rétro-propagation

Apprentissage perceptron

- ◆ But : Trouver un ensemble de poids qui minimise les erreurs sur un ensemble d'entraînement
- ◆ Signaux d'entrée = -1 ou 1
- ◆ Fonction d'activation : seuil
 - 1 si $\sum W_i X_i \geq \text{Seuil}$
 - 1 si $\sum W_i X_i < \text{Seuil}$

Apprentissage perceptron

- ◆ Apprentissage supervisé :
 - comparaison de la solution avec celle d'un expert
 - réduction des erreurs par un ajustement des poids
- ◆ Utilisation pour classifier (sortie du réseau = classe)

Applications

- ◆ Classification
- ◆ Reconnaissance de formes
- ◆ Prédiction
- ◆ Filtration (bruit)
- ◆ ...

Exemples w w w

- ◆ <http://suhep.phy.syr.edu/courses/modules/MM/sim/perceptron.html>
- ◆ <http://diwww.epfl.ch/mantra/tutorial/french/perceptron/html/index.html>

3) Approche sociale et émergente



(c) regentsprep.org

© Capus, Potvin et Tourigny, 2003

45

3) Approche sociale et émergente



- ◆ Basée sur l'observation des phénomènes d'équilibre des mondes humain et animal
- ◆ Recherche sur les algorithmes génétiques, la programmation génétique et la vie artificielle
- ◆ Systèmes de classifieurs basés sur les algorithmes génétiques ⇒ processus d'apprentissage automatique

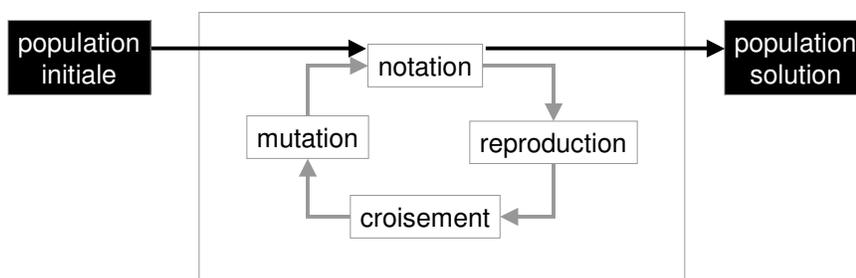
© Capus, Potvin et Tourigny, 2003

46

Structure d'un algorithme génétique

- ◆ Utilisation des principes du néodarwinisme
- ◆ Exploration d'un espace de solutions à partir d'une population initiale d'éléments jusqu'à l'apparition de la solution au problème posé
- ◆ Représentation par des chaînes binaires

Algorithme génétique : cycle d'évolution



Algorithme génétique : cycle d'évolution

- ◆ Création aléatoire d'une population d'éléments représentant les premières solutions envisagées au problème posé (génération 0)
- ◆ Application d'une série d'opérateurs génétiques (reproduction, croisement et mutation)
- ◆ Notation des individus de chaque génération grâce à une fonction d'évaluation

Algorithme génétique : cycle d'évolution

- ◆ Sélection des meilleures solutions pour transmettre leur patrimoine génétique
- ⇒ Concentration de la population vers une solution correspond à un optimum global de la fonction d'évaluation

Algorithme génétique : opérations

- ◆ **Création** : choix aléatoire de la première génération
- ◆ **Évaluation** : notation de chaque individu en fonction de son aptitude à résoudre le problème posé (différenciation des individus pour reproduire les meilleurs et éliminer les moins efficaces)

Algorithme génétique : opérations

- ◆ **Reproduction** (version artificielle de la sélection Darwinienne) : sélection d'un certain nombre d'individus de la population $P(t)$ pour créer la génération suivante $P(t+1)$ en fonction de leur note pour permettre d'augmenter leur chance de transmettre leur patrimoine génétique

Algorithme génétique : opérations

- ◆ **Roulette** : utilisation d'une roue de tirage aléatoire sur laquelle chaque individu est représenté par un secteur proportionnel à sa note
- ◆ **Tournoi** : tirage aléatoire de n participants et le vainqueur est celui qui a la meilleure note parmi les n participants

Algorithme génétique : opérations

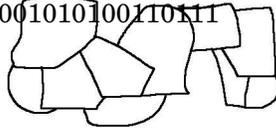
- ◆ **Croisement** : mélange de deux individus issus de la reproduction pour en produire de nouveaux
- ◆ **Mutation** : apparition d'une perturbation au niveau du code de l'individu
- ◆ **Élitisme** : Le meilleur individu de la population est conservé

Algorithme génétique : exemple

◆ Exemple d'encodage d'un individu qui consisterait au coloriage d'une carte sans avoir la même couleur sur 2 sections conjointes. On veut utiliser le minimum de couleurs possibles.

- 10 sections et nous avons 8 couleurs possibles
- Chaque couleur peut être représentée sur 3 bits
- Donc un individu aura $10 * 3$ bits de long = 30 bits
- Un individu pourrait ressembler à :

011001010010101001010100110111



Algorithme génétique : exemple

◆ À l'origine nous devons nous générer une population aléatoire d'individus

- Chacun des bits est choisi aléatoirement
- Il ne reste plus qu'à faire évoluer la population pour obtenir notre réponse

Algorithme génétique : exemple

◆ Croisement

- On choisit 2 individus de l'ancienne génération
- On choisit un point de croisement sur l'individu
- On interchange les 2 extrémités une fois le point de croisement passé

Parent 1	0110 000110
Parent 2	1100 111100
Enfant 1	0110 111100
Enfant 2	1100 000110

© Capus, Potvin et Tourigny, 2003

57

Algorithme génétique : exemple

◆ Mutation

- Avec une certaine probabilité préalablement fixée, un bit de l'individu pourrait changer

Enfant 1	0110111100
Enfant 2	1100000110
Enfant muté 1	0010111100
Enfant muté 2	1101000100

© Capus, Potvin et Tourigny, 2003

58

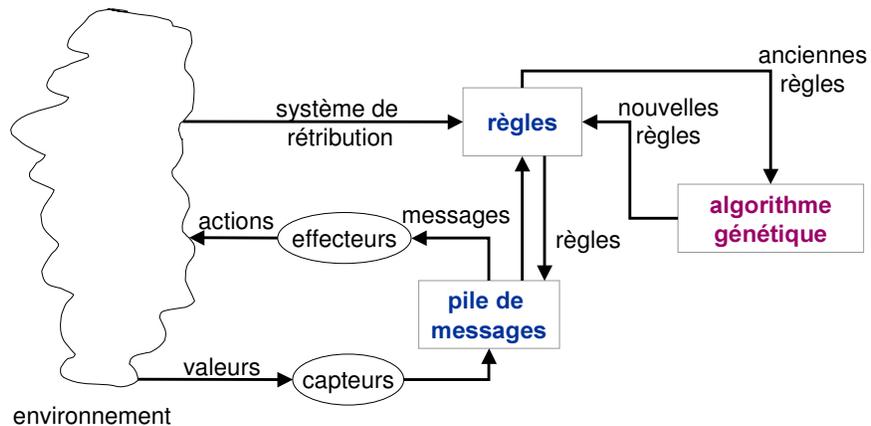
Algorithme génétique : exemple www

◆ http://cs.felk.cvut.cz/~xobitko/ga/example_f.html

Vie artificielle

- ◆ Automates cellulaires
- ◆ Règles simples déterminant la survie en fonction des cellules voisines

Systèmes de classifieurs



© Capus, Potvin et Tourigny, 2003

61

Systèmes de classifieurs

- ◆ Codage des informations : un caractère spécial # (peu importe)
- ◆ Adaptation des algorithmes génétiques classiques pour ne pas renouveler complètement la population

© Capus, Potvin et Tourigny, 2003

62

Systemes de classifieurs

- ◆ Ajout de 2 paramètres :
 - proportion de nouvelles règles à remplacer
 - fréquence d'appel (fixée ou adaptative)
- ◆ Utilisation des opérateurs génétiques classiques :
 - sélection grâce à la roulette
 - mutation adaptée à l'alphabet ternaire {0, 1, #}

3) Approche sociale et émergente

- ◆ Développer une solution avec l'approche sociale et émergente : empirisme
 - Spécifier les paramètres de départ
 - Vérifier le résultat avec ces paramètres
 - Changer un ou plusieurs paramètres et recommencer si la solution n'est pas atteinte
- ◆ Applications
 - Agents et systèmes multi-agents
 - Problèmes où il faut tenir compte de beaucoup de facteurs