Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider Biskra

Faculté des sciences exactes, de la science naturelle et de la vie

Département d'Informatique

N°d'ordre :……
Série :……

MÉMOIRE

Présenté en vue de l'obtention du diplôme de Magister en Informatique

Option**: Intelligence Artificielle**

Présenté par :
M^{elle} Asma BENDAHMENE

UNE APPROCHE BASÉE ALGORITHME GÉNÉTIQUE POUR L'APPRENTISSAGE CHEZ UN AGENT

Soutenu le : / / 2011 devant le jury composé de :

Mr. Melkemi Kamel Eddine	**M.C. Université de Biskra**	**Président**
Mr. Kazar Okba	**M.C. Université de Biskra**	**Rapporteur**
Mr. Laouar Ridha	**M.C. Université de Tebessa**	**Examinateur**
Mr. Khelil Nacer	**M.C. Université de Biskra**	**Examinateur**

À la mémoire de ceux que

mon cœur aime …

Remerciement

La réalisation de ce travail est une tranche de vie à part entière qui s'est nourrie de nombreuses et diverses influences. Qu'ELLAH le tout puissant reçoive ma gratitude pour m'avoir donné la force morale et physique pour l'achever.

Beaucoup de personnes ont participé directement ou indirectement à sa concrétisation et méritent d'être remerciées, quel que soit, le degré de leur investissement. Malheureusement, tout le monde ne peut apparaître nommément ici et je me dois de faire une sélection subjective.

Je tiens d'exprimer toutes ma reconnaissance auprès de mon encadrant Okba Kazar. Je le remercie pour son immense patience, sa disponibilité, sa grande implication ainsi que pour son énorme soutien.

Je remercie également mes membres de jury, Mr Melkemi Kamel Eddine, Khelil Nacer et Laouar Mohammed Ridha qu'ils m'ont accordé en acceptant d'évaluer mon travail.

Ma gratitude va particulièrement à mon enseignant Mr Khaled Rezag, pour l'enthousiasme qu'il a manifesté à l'égard de mon mémoire. Il m'a prodigué des conseils inestimables avec son aide précieuse.

Je souhaite aussi remercier Melle Sihem Slatnia, pour toutes les informations qu'elle m'a données. elle m'aura en effet permis de mieux comprendre les enjeux de ce mémoire, et d'être plus à même d'interpréter mes résultats.

Je remercie Mr Angel Fernando Morales Kuri, docteur en informatique, université Western Kennedy, États-Unis, pour son aide, sa sympathie et sa disponibilité à me répondre à tout moment.

Pour finir, mes remerciements vont sortir pour partie du cadre professionnel pour aller : vers les membres de ma petite et grande famille pour leur soutien indéfectible et leurs encouragement. Mes remerciements s'adressent aussi à mes amies qui ont partagé mon quotidien et d'avoir été là lorsque j'en avais besoin.

À ceux qui auraient été très fiers de me voir achever ce mémoire.

RÉSUMÉ

L'agent est muni d'un ensemble de tâches à effectuer et il est appelé à les exécuter, cependant ses propres connaissances et compétences ne sont pas suffisantes, ou bien tellement son environnement est dynamique, il doit actualiser son état interne d'une façon permanente pour survivre et atteindre son but.

L'apprentissage est alors, une caractéristique fondamentale afin que l'agent puisse augmenter ses connaissances et améliorer ses performances en utilisant ses expériences. L'une des techniques d'apprentissage est les algorithmes génétiques qui sont favorables dans tel domaine à cause de leurs caractéristiques.

L'objectif de notre travail consiste à élaborer une architecture d'un agent capable d'apprendre en utilisant un algorithme génétique incorporé avec un mécanisme d'apprentissage par renforcement.

Mots clés : **Apprentissage artificiel, Algorithme génétique, Apprentissage par renforcement, Agent.**

ABSTRACT

Along its life, the agent has a set of tasks to do, however, its own knowledge and competences are not sufficient, and its environnement is so dynamic, it must update its local state in ordre to survive and acheive its goal.

Learning is then a key feature to enable the agent to increase knowledge and improve performance by using past experiences and therfore effectively acts and reacts in its environnement.Genetic algorithms are more suggested in such demand, because of their evolutionary character.

Our work consists in proposing a model based genetic algorithm incorporated with mechanism of rienforcement learning for the agent learning.

Keywords: **Artificiel learning, Genetic algorithm, Reinforcement learning, Agent.**

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Liste des équations

Acronyme

ACL	Acts Communication Language
AD	Arbre de Décision
AG	Algorithme Génétique
AMS	Agent Management System
AR	Apprentissage par Renforcement
DF	Director Facilitator
GUI	Graphic User Interface
KPPV	K-Plus Proches Voisins
MMC	Modèles de Markov Cachées
NASDAQ	National Association of Dealers Automated Quotes System
NYSE	New York Stock Exchange
Pc	Probabilité de croisement
Pm	Probabilité de mutation
RB	Réseaux Bayésien
RMA	Remote Management Agent
RN	Réseaux de Neurones
SMA	Système Multi Agents
UML	Unified Modeling Language

Introduction Générale

« L'important de la pédagogie n'est pas d'apporter des révélations, mais de mettre sur la voie. »

Un même mystère, Pierre Dehaye

 epuis sa naissance, l'intelligence artificielle cherche de construire des systèmes informatiques qui égalisent voir dépasseraient l'homme dans nombreuses activités réputées intelligentes comme raisonner ou résoudre des problèmes complexe.

Les branches d'intelligence artificielle sont réparties en deux catégories : les problèmes et les techniques. Les problèmes sont plutôt des domaines d'application alors que les techniques se veulent plus des méthodes abstraites et génériques.

L'apprentissage est l'une des caractéristiques de l'intelligence humaine et fait partie du processus cognitif qu'est la mémoire. Cette mémoire représente la faculté de conserver des informations qui sont constituées par des expériences, des connaissances et elle permet de se souvenir des apprentissages antérieurs. Apprendre signifie donc intégrer de nouvelles informations en vue de s'en servir dans le futur. Ces connaissances sont extraites d'un ensemble d'exemples ou des propres expériences. Lorsqu'une nouvelle information apparaît, le système peut la classifier, faire une prédiction à partir d'exemples, ou la généraliser à des nouvelles situations.

L'intelligence fait intervenir le concept de compréhension, non pas qu'un système dispose plus de connaissances, mais surtout comment les exploiter pour qu'il s'améliore progressivement.

Les systèmes multi agents, composés d'un ensemble d'agent, sont l'une des dernières générations de systèmes informatiques intelligents.

Un agent est une entité logicielle ou matérielle situé dans un environnement, menu d'une tâche à résoudre et un objectif à atteindre.

Pour l'accomplissement de son but, l'agent nécessite des connaissances et des compétences qui le permettent à réagir face aux différentes situations rencontrées. Il arrive que ces prérequis ne sont pas disponibles au début de l'exécution pour deux raisons : l'environnement est dynamique, il subit des changements rapides et vastes, ou bien l'environnement lui-même n'est pas bien déterminé dés le début. Alors l'apprentissage semble être une faculté indispensable tant pour l'amélioration des performances d'agent que pour lui permettre à disposer même si les connaissances initiales sont modiques.

Lors de la conception d'un système d'apprentissage plusieurs questions viennent tout naturellement à l'esprit, surtout si le système évolue dans un environnement dynamique.

- Comment construire un système d'apprentissage capable d'agir de manière autonome dans un système complexe ?
- Comment exploiter les connaissances de l'apprentissage pour améliorer les performances du système ?

De plus, on identifie plusieurs conditions nécessaires à la conception d'un système d'apprentissage, on peut les résumer de la façon suivante :

- apprendre à s'adapter rapidement aux environnements changeants et inconnus ;
- assurer une certaine qualité des meilleures solutions trouvées ;
- converger rapidement vers une solution ;
- s'adapter à l'environnement pour permettre un apprentissage optimal ;
- garantir la fiabilité du système dans le temps ;
- extraire des connaissances à partir des informations ;
- prévoir le cas de contextes non prévus ;
- proposer des actions qui sont le résultat de ses simulations les plus prometteuses ;
- assurer une possibilité de correction des paramètres et de retour si l'information est disponible.

Cependant une grande variété des techniques d'apprentissage ont été développés tel que les algorithmes génétiques et l'apprentissage par renforcement. Chacune est caractérisée par des propriétés distinctes.

Apprendre par renforcement, c'est apprendre par interaction avec l'environnement de sorte à maximiser un certain signale de récompenses. Un agent apprenant par renforcement progresse par essais et erreurs. L'un des problèmes d'apprentissage par renforcement est le compromis entre l'exploration et l'exploitation. Pour obtenir un maximum de récompenses, un agent apprenant par renforcement préférera des actions qu'il a essayé dans le passé et qui sont avérées efficacement pour obtenir des récompenses. Néanmoins, pour découvrir de telles actions, il aura dû accomplir des actions qu'il ne connaissait pas à priori. L'agent doit exploiter ce qu'il connaît déjà afin d'obtenir des récompenses, mais il doit également explorer de sorte à faire des choix encore meilleurs dans le futur. Ce dilemme exploration/exploitation est résolu par la combinaison d'algorithme d'apprentissage par renforcement avec un algorithme génétique.

L'usage des algorithmes génétiques convient parfaitement pour la réalisation d'un système d'apprentissage car il est un concept basé sur la théorie de l'évolution des espèces et comprend par le fait même des aptitudes d'adaptation très fortes.

Leur principe se dérive de la théorie darwinienne qui consiste à faire évoluer une population des individus parents pour obtenir des fils, à l'aide des opérateurs génétiques ; croisement et mutation. Chaque individu code une solution au problème abordé, sa pertinence à la résolution est mesurée par une fonction d'adaptabilité dite fitness. Cette procédure permet de passer d'une génération à une autre plus évolues jusqu'à arriver à une solution optimale. L'algorithme génétique assure à la fois l'exploration d'un espace de recherche et l'exploitation des solutions déjà explorées.

Les systèmes d'apprentissage représentent un enjeu important, et ce pour différents domaines comme les marchés financiers. En effet les marchés financiers proviennent du système ouvert dont ils changent dans le temps et très souvent un sujet d'importance variation dues aux informations extérieures qui produisent des résultats non anticipés. Les agents intervenant doivent être en mesure à réagir efficacement aux fluctuations rencontrées. Différentes théories ont été développées tant pour la modélisation économique de la résolution des problèmes complexes que pour la restitution du comportement interne des

agents et des mécanismes de prise de décision. L'approche évolutionnaire dont le principe est de considérer des agents qui ont initialement une très petite rationalité et connaissances spécialisées de leur domaine d'action. Ensuite ils sont amenés à s'adapter et à apprendre, et devenant petit à petit experts dans leur propre domaine. C'est à l'apprentissage alors de jouer un rôle primordial dans l'amélioration des décisions d'un agent face aux différentes situations.

Nous présentons une approche d'apprentissage, répondant à ces besoins, à travers ce mémoire organisé en quatre chapitres :

Le premier chapitre sert à présenter un état de l'art sur l'apprentissage automatique, en mettant l'accent sur les techniques et les caractéristiques d'un apprentissage machine, spécifiquement chez un agent. Des travaux d'apprentissage sont ainsi cités.

Ensuite, le deuxième chapitre est destiné à décrire notre modèle proposé avec une description détaillée et complète des algorithmes développés.

Pour la validation de notre approche, nous exposons dans le chapitre 3 un cas d'étude sur les marchés financiers précisément la bourse NASDAQ, avec une projection des paramètres des algorithmes d'apprentissage sur ce cas.

Dans le chapitre 4, nous présentons les résultats produits par notre application avec une analyse des paramètres algorithmiques choisis et leur effet sur le processus d'apprentissage.

Enfin, nous concluons ce manuscrit par une synthèse de notre contribution, une analyse critique de notre approche, et les améliorations et travaux en perspective.

Chapitre 1

Revue de la littérature

En informatique, l'expert donne une méthode pour résoudre un problème alors qu'en « machine learning », l'expert donne une méthode pour 'apprendre' à résoudre (automatiquement) une classe de problèmes.

Un programme d'ordinateur qui peut apprendre par l'expérience pour faire une tâche particulière en respectant certains critères ! Quelle idée ambitieuse. Pourtant de nombreuses applications ont déjà vu le jour, le « machine learning » est une manière originale et récente de voir les sciences : les sciences nous aident à comprendre notre environnement, pourquoi ne joueraient-elles pas le même rôle pour les ordinateurs ?

Ce chapitre est consacré à un état de l'art sur l'apprentissage en intelligence artificielle. Nous le présentons en trois sections.

D'abord, nous commençons par aborder le phénomène d'apprentissage automatique, mettant en circonstance, l'ensemble de concepts et techniques (algorithme génétique et apprentissage par renforcement) avec une catégorisation selon laquelle on décrit les principes importants d'un apprentissage machine.

La section suivante, prend en étude le concept agent, en assignant les différentes définitions et propriétés lui associées. Ainsi que l'organisation dite système multi agents et l'intérêt de l'apprentissage au sein d'un agent indépendamment ou bien en collaboration (interaction) avec les autres.

Nous finissons par une section réservée à la découverte les travaux relatifs.

1.1 Théorie de l'apprentissage

1.1.1 Définition

Le rêve des chercheurs, à développer des machines qualifiées intelligentes dont tous les êtres humains ont l'apanage, a conduit pour imiter leur manière d'agitation. La caractéristique d'apprendre offre l'être humain l'intelligence et par conséquent les machines aussi doivent devenir apte à apprendre. La faculté d'apprendre repose sur un processus de mémorisation des informations qui sont constituées par des expériences et des connaissances. Extraire de cette masse d'information celles qui sont pertinentes dans un but explicatif ou décisionnel présente l'un des objectifs d'apprentissage automatiques. Plusieurs définitions ont été proposées. ***Mishalki*** a définit l'apprentissage comme suit ***[25,77]***:

« L'apprentissage est la construction de nouvelles connaissances ou amélioration de connaissances déjà existantes afin d'améliorer les performances du système »

Apprendre signifie l'intégration de nouvelles informations en vue de s'en servir dans le futur pour aider à résoudre un problème particulier.

Apprendre c'est aussi raisonner : découvrir des analogies et des similarités, généraliser ou particulier une expérience, faire une classification ou prédiction, tirer parti des échecs et erreurs passés pour des raisonnements ultérieurs ***[52].***

En résumé, l'apprentissage est un terme très général qui décrit le processus selon lequel la machine peut accroître ses connaissances et améliorer ses performances au cours du temps. Comme le définit ***Simon [87,101]***, un pionnier dans les sciences cognitives et intelligence artificielle :

« Learning denotes changes in the system that are adaptative in the sense that they enable the system to do the same task or tasks downs from the same population more effectively the next time »

Les experts considèrent qu'il est plus simple de décrire des exemples ou des cas expérimentaux plutôt que d'expliciter des processus de prise de décision ***[52]***. Alors ils ont construisent des systèmes d'apprentissage entrainés par des exemples pour traiter les nouveaux acquis.

1.1.2 Formulation d'apprentissage

Un algorithme d'apprentissage A est un algorithme qui a pour fonction d'apprendre à effectuer une tâche à partir d'un ensemble S de données (exemples). Chaque donnée Z_i est constitue d'un objet d'entrée x_i et une valeur de sortie y_i

$$Z_i = (x_i, y_i)$$

$$S = \{Z_i\}_{i=1}^{m} = \{Z_1, Z_2 \dots \dots \dots \dots Z_m\}$$

L'objectif de l'algorithme d'apprentissage est de construire pour tout couple (x, y) une bonne fonction h(x) appelé hypothèse qui représente la solution du problème en discussion, tel que h(x) = y.

L'entrée d'algorithme d'apprentissage A est un ensemble S et la sortie est une fonction appelée h.

On écrit $A(S) = h$

1.1.3 Caractéristiques d'apprentissage

Parmi les caractéristiques primordiales de l'apprentissage, nous situons les trois suivantes ***[15]***.

- **a.** l'abstraction des données ; savoir si la méthode utilisé engendre des données proches de grandeur physique ou plutôt de symbole.
- **b.** l'élément temporel ; il est important de déterminer si le système apprenant est en mesure de s'adapter « en ligne » ou il considère de façon différé toutes nouvelles modifications de l'environnement.
- **c.** le rôle de l'enseignant ; il s'agit de savoir si le système apprend de façon autonome ou il a besoin d'être supervisé.

1.1.4 Conditions d'apprentissage

Pour toutes formes d'apprentissage, un ensemble de conditions sont nécessaires pour assurer le progrès adéquat du système apprenant ***[14]***.

Condition 1 :

« Toute forme d'apprentissage nécessite la répétition des décisions dans le temps. »

La première condition consiste à permettre au système en phase d'apprentissage d'effectuer plusieurs essais, car tant qu'il y a plus de répétition, il y aura plus d'expériences.

Condition 2 :

« Toute forme d'apprentissage nécessite un mécanisme de rétroaction environnemental. »

Afin qu'un système abouti à un apprentissage, il faut qu'il reçoit un feedback (rétroaction) de son environnement à la suite de ses propres décisions et les décisions des autres.

Condition 3:

Toute forme d'apprentissage nécessite d'un mécanisme d'adaptation des décisions.

Selon la troisième condition, non seulement l'acquisition de nouvelle information par le système apprenant mène à un apprentissage, mais aussi le renseignement sur lequel ces informations peuvent être utilisées sont aussi nécessaires.

Condition 4 :

Toute forme d'apprentissage nécessite l'existence un mécanisme de stockage de l'information ; la mémoire.

En effet, s'il n'y a pas une sauvegarde de conséquences obtenues dans le passé, le système ne bénéficie pas de ses décisions antérieures. Le stockage d'informations offre au système la possibilité de modifier sa décision sur la base de l'expérience passée et d'améliorer ses performances. En revanche, la mémorisation doit être guidée par des stratégies adéquates en tenant compte l'espace mémoire comme un facteur critique.

1.1.5 Classification d'apprentissage automatique

Dans la littérature on dénombre une grande variété de forme d'apprentissage. Il y a cependant différents critères importants pour classifier ces méthodes, dans ce qui suit un survol sur les plus récurrents.

1.1.5.1 Selon les connaissances manipulées

La représentation des connaissances et le raisonnement à partir de ces représentations a donné naissance aux deux modèles.

- ★ **Apprentissage symbolique :** l'apprentissage symbolique manipule des connaissances qualitatives sous forme de symbole. Il consiste à élaborer des méthodes permettant d'extraire des connaissances structurelles à partir d'instances peu structurées. L'avantage principal de l'apprentissage symbolique est sa portée sémantique forte. Un expert qui analyse le système apprenant peut comprendre la façon dont celui-ci fonctionne et les résultats fournis sont facilement interprétables ***[24, 93, 116]***.
- ★ **Apprentissage numérique :** on fait appel à ce type d'apprentissage principalement pour traiter des connaissances numériques quantitatives. L'apprentissage numérique utilise des techniques statistiques, se révèlent d'être portables et permettent une grande adaptabilité, car leurs non dépendant de symboles. Par contre, le fonctionnement interne du système est opaque. Il est très difficile voir impossible de comprendre comment le système apprend ***[40, 97]***.

1.1.5.2 Selon le type d'information disponible

En apprentissage, l'existence ou non d'un enseignant ainsi que l'information fournit par ce dernier fait distinction entre trois classes.

- ★ **Apprentissage supervisé :** ce genre, nécessite un expert (enseignant) qui fournit des exemples formés de situations et actions associées afin de contrôler la réponse obtenue par le système et le diriger pour améliorer ses performances. L'apprentissage supervisé consiste à réaliser un saut inductif en passant des exemples particuliers à des connaissances générales. Le but est de généraliser l'association apprise à des situations nouvelles ***[17, 31, 55, 101]***.
- ★ **Apprentissage par renforcement :** par opposition à l'apprentissage supervisé l'expert ici a le rôle d'évaluateur et non plus instructeur. Il fournit une mesure

(critique) sur le résultat généré indiquant si l'action réalisée est appropriée ou non. En fonction de récompenses ou punitions le système décide lui-même s'il doit modifier ou non son comportement ***[34, 75]***.

1.1.5.3 Apprentissage non supervisé: dans ce cas, le système n'exige pas la présence d'un expert et ne dispose d'aucune information extérieure concernant les résultats attendus ou les mesures de critique. Il découvert lui-même de nouvelles connaissances suite à la recherche des corrélations existantes entre les exemples d'entrainement en entrée, puis le regroupement des données semblables en clusters ***[71, 82]***.

1.1.5.4 Selon l'objectif attendu du processus d'apprentissage

Une autre classification proposée qui distingue entre trois types classés selon l'objectif attendu du processus d'apprentissage.

Dans chaque type, le résultat final (solution) supposé connu d'avance, la différence se résume dans la méthode appliquée pour atteindre la solution désirée ; par évaluation, optimisation ou bien un entraînement.

- ★ **Apprentissage par évaluation:** consiste à déterminer pour un problème donné et un ensemble de méthodes de résolutions possibles, quelle est celle qui convient le mieux à la solution ***[47]***.
- ★ **Apprentissage par optimisation:** l'objectif dressé par ce type est de déterminer la méthode de résolution d'un problème qui a donné lieu à la solution. Sachant que cette méthode doit être optimal (vérifier un critère d'optimalité) ***[23, 24,47]***.
- ★ **Apprentissage par entrainement :** consiste à adapter les connaissances actuelles d'un problème donné afin de maximiser la probabilité d'avoir la solution attendue ***[47]***.

1.1.5.5 Selon la stratégie utilisée :

L'apprentissage peut être encore envisagé selon quatre stratégies adoptées pour qu'un système apprendre qui sont les suivantes :

Apprentissage par cœur : consiste en l'acquisition de connaissances et d'aptitudes sans recourir à une modification du système apprenant ou des inférences de sa part. Il existe une absence de généralisation, les connaissances mémorisées ne peuvent être réutilisées dans

des nouvelles situations, le résultat stocké qui convient le mieux à la nouvelle situation est fourni sans aucune modification ***[77, 78]***.

- ★ **Apprentissage par instruction :** consiste en l'acquisition de connaissances et d'aptitudes qui nécessitent l'intégration des nouvelles représentations de son environnement. L'apprentissage est caractérisé par une forme locale de généralisation. Ces instructions sont mémorisées et utilisées si nécessaire dans le contexte qui leur est propre pour résoudre certaines tâches ***[8]***.
- ★ **Apprentissage par analogie :** l'apprentissage par analogie désigne les méthodes dites paresseuses ; c'est à dire n'effectuant pas de généralisation des données disponibles. Il consiste en l'acquisition de connaissances pour les stocker comme référence de cas. Ce mécanisme permet au système de générer un résultat à l'aide des situations plus familières adaptées à la nouvelle ou bien adopter une attitude dans une situation inconnue ***[10, 24, 52, 108]***.
- ★ **Apprentissage par exemple :** consiste en l'acquisition de connaissances pour la caractérisation et la discrimination d'un concept. Ce genre d'apprentissage se caractérise par généralisation globale qu'il nécessite. Les exemples initiaux sont oubliés et une nouvelle connaissance générale comprime ces exemples est mémorisée ***[77]***.

1.1.6 Les paradigmes d'apprentissage

Généralement, un paradigme se caractérise par : un modèle, le plus souvent paramétrique, une façon d'interagir avec l'environnement, une fonction de coût à minimiser, un algorithme pour adapter le modèle en utilisant les données issues de l'environnement, de façon à optimiser la fonction de coût. Nous allons détailler, au suivant, celles les plus fréquents.

1.1.6.1 Les réseaux de neurones(RN)

On appelle RN un ensemble de calculateurs numériques qui agissent comme des unités élémentaires, ils sont reliés entre eux par un ensemble d'interactions pondérées qui transmettent des informations numériques d'un neurone formel à l'autre.

L'apprentissage du RN consiste alors à modifier répétitivement les valeurs de pondérations des interactions jusqu'à ce que la performance du réseau atteigne le niveau désiré. ***[31, 121]***

Cet apprentissage se fait suivant deux approches**.** Soit on fixe l'architecture et on apprend les poids comme le fait la règle de hebb, la règle de windroff et l'algorithme de rétro propagation. Soit on apprend de manière constructive le nombre d'unités et les poids. ***[55, 113]***

En effet, les RN ont été appliqués avec succès à l'apprentissage de tâches de classification et d'approximation de fonctions. Leur utilisation permet de passer des données au prédiquant, sans intermédiaire, sans codage et sans interprétation sujette à caution. Ils possèdent également une grande résistance au bruit ou au manque de fiabilité des données ***[9].***

Néanmoins, ils présentent des inconvénients. Ils ne dispensent pas de bien connaître son problème, de définir des classes aux pertinences sans oublier des variables importantes. D'autre part un RN est une boite noire qui n'explique pas ses décisions dont les résultats obtenus seront difficilement interprétables. Ainsi que l'ajustement du paramètres (poids, unités) devenu de plus en plus une tâche ardue autant que les problèmes sont complexes, et par conséquent le réseau atteint un état de sur apprentissage, et perd ses pouvoirs à résoudre les problèmes en discussion ***[113].***

1.1.6.2 Les arbres de décisions(AD)

Les AD représentent l'une des techniques les plus connues et les plus manipulées en classification. Ils permettent d'inférer des conclusions générales à partir des exemples.

Le processus de prédiction de classe (concept ou cas) est utilisé principalement dans le cadre d'apprentissage supervisé.

L'apprentissage d'un AD consiste à la construction de cet arbre, plusieurs méthodes ont été proposées (ID3, CHAID, C5…). Leurs principe sert à trouver pour chaque nœud un partitionnement des individus d'entraînement que l'on représente sous la forme d'un arbre dont les nœuds internes désignent des tests sur les attributs des données en entrée et dont les feuilles indiquent des correspondants aux données ***[23, 91].***

Les AD ont prouvé leur qualification au domaine d'apprentissage automatique. Leur succès est notamment dû à leur aptitude à traiter des problèmes complexes de classification. En effet, ils offrent une représentation facile à comprendre et à interpréter ainsi qu'il possèdent une capacité à produire des règles logiques de classification. Leur rapidité en classement dépond tout simplement du nombre majeur des nœuds en partant de la racine aux feuilles ***[7]***.

Auprès de tous les points forts cités juste avant les AD présentent quelques désavantages. Entre autre, lorsque l'on construit un arbre de décisions on risque ce que l'on appelle un su rajustement du modèle ; Le modèle semble parfait mais en réalité il n'est plus et il sera indispensable de chercher un autre arbre plus petit ayant la plus grande performance possible ce qui nécessite une phase d'élagage d'arbre par issu un temps important est consommé. Leur performance est diminuée quand il y a beaucoup de classes. D'autre part ils sont sensibles à la fois au manque de données (data overfitting) et aux données de valeurs numériques ***[7]***.

1.1.6.3 les réseaux bayésiens(RB)

Les RB sont connus comme un formalisme apte à assurer la prise en compte de l'incertain en modélisation des problèmes. ***[80]***

La structure de ce type de réseau est simple : un graphe dont lequel les nœuds représentent des variables aléatoires correspondant aux données et les arcs reliant ces derniers sont attachés à des probabilités conditionnelles. Ils exploitent le théorème de bayes★ afin d'apporter des solutions. ***[84]***

L'apprentissage d'un RB peut être réalisé selon deux manières. La première fait un apprentissage des paramètres où on suppose que la structure du réseau à été fixé et il faudra estimer les probabilités conditionnelles de chaque nœud du réseau. Tandis que la deuxième sert à apprendre la structure; le but est de trouver le meilleur graphe représentant les tâches à résoudre. ***[72, 85, 86]***

À la suite de leur représentation probabiliste sous forme d'un graphe, les RB offrent un mécanisme efficace de résolution du problème. Ainsi que leur pouvoir de gérer des données incomplètes, comme ils permettent d'apprendre la relation causale qui peut aider la prise de décision. De plus, ils donnent la possibilité de rassembler des connaissances de diverses natures dans un même modèle. ***[53, 56, 84]***

En revanche, les RB ont pas mal d'inconvénients surtout la complexité des algorithmes. La plus parts des algorithmes développés pour l'inférence et l'apprentissage utilisent des variables discrètes et la manipulation des variables continues est en cours de

★ Le nom de réseau bayésien provient de la formule d'inversion de Bayes : $P(H|e) = P(e|H) \times P(H)/P(e)$ ***[84].***

recherche. Un autre défi est que la généralité du formalisme des RB aussi bien en terme de représentation que d'utilisation les nœuds les rend difficile à manipuler à partir d'un certain taille ***[85]***.

1.1.6.4 Les k-plus proches voisions(Kppv)

C'est une approche dite « memory-based », elle suit un raisonnement par analogie, utilisée principalement aux problèmes de classification.

Cette méthode diffère des autres méthodes d'apprentissage car aucun modèle n'est induit à partir d'exemple. Les données restent telles quelles ; elles sont stockées en mémoire. Quand on parle de voisin cela implique la notion de distance ou de dissemblance. La décision consiste à chercher les k échantillons les plus proches de l'observation et d'affecter la classe la plus fréquentée dans ces K instances. Différents outils statistiques sont utilisés, la plus populaire est la distance eucleudienne ***[19, 70, 100]***.

Les expériences menées avec les Kppv montrent qu'ils résistent bien aux données bruitées, ainsi qu'ils présentent l'avantage de la simplicité ***[22***, ***32]*.** Par contre ils requièrent de nombreux exemples, ce qui rend le processus de décision complexe. Bien que le temps d'apprentissage soit inexistant, la classification d'un nouveau cas est coûteuse puisqu'il faut comparer ce cas à tous les exemples déjà classés ***[4, 51].***

1.1.6.5 les machines à vecteurs support (SVM)

Les SVM sont une méthode de classification qui montre de performance dans la résolution de problèmes variés tel que la classification et la régression. Ils emploient un apprentissage supervisé dont le principe sert à définir un séparateur linéaire ou non entre deux ensembles de points (données d'apprentissage) par un hyperplan.

L'idée est de maximiser la distance appelé marge entre l'hyperplan séparateur et les exemples. Les instances les plus proches qui suffisent à déterminer cet hyperplan sont appelés vecteurs de support. ***[11, 26, 50, 74].***

Les SVM introduisent des aspects positifs à savoir leur adaptabilité aux problèmes non linéairement séparables. Dans les SVM peu de paramètres à fixer et aucun ajustement manuel est requis car ils ont l'habilité de trouver automatiquement des paramètres adéquats. Ainsi qu'ils garantissent un optimum global unique et rehaussent une aptitude forte de généralisation ***[27, 66***, ***107, 108]***.

En revanche, les SVM ne sont pas tolèrent au bruit et aux outliers ***[108]***.

1.1.6.6 Les Modèles de Markov Cachées (MMC)

Les MMC sont une méthode statistique qui modélise des séquences d'états, nommés états cachés car non observables. Le modèle inclut des probabilités de transitions entre ces états afin de modéliser les observations. Ils permettent la représentation des processus stochastiques ***[43, 63].***

L'objectif de faire apprendre un MMC est que cette dernière ait la plus forte probabilité possible de générer ces observations. Cela se fait par l'ajustement des différents éléments du MMC (nombre d'états, probabilité de transition) ***[2, 92, 106].***

Les MMC sont performants quand il s'agit d'opposer un alignement temporel. Néanmoins elles présentent quelques insuffisances tel que le choix de l'architecture du MMC ; il n'existe pas une méthode universelle pour définir le nombre d'états optimal or cette valeur est très influente sur la qualité de la solution finale ***[43, 102]***. Pendant l'étape d'apprentissage, il arrive que les MMC se trouvent au point de risque à converger vers un minimum local au lieu d'un minimum global, ce qui avachi le résultat obtenu. Une autre limite à citer concerne le passage à l'échelle, l'impossibilité de la prise en compte efficace de l'influence de phénomènes indépendant et la difficulté de généralisation ***[3]***.

1.1.6.7 Les algorithmes génétiques(AG)

Les AG ont été mis au point par Holland dans les années 60 aux États-Unis ***[57]***. Ils sont ensuite raffinés et popularisés par De Jong ***[28]***, Grefenstette ***[49]*** et Goldberg **[44]**.

Son principe s'inspire des mécanismes de l'évolution biologique pour les transporter à la recherche de solutions adaptées au problème qu'on cherche à résoudre. Ils reprennent les notions de la génétique du vivant et exploitent les éléments suivants:

- **Gène :** la plus petite entité existante symbolisé par un caractère, non forcement entier.
- **Individu/chromosome :** une séquence de gènes représentant une solution potentiel du problème correspondant à une valeur codée de la variable (ou des variables) en considération (génotype).

- **Population :** un ensemble de chromosomes ou des éléments de l'espace de recherche, donc des valeurs codées des variables (phénotype).
- **Génération :** ensemble d'individus de la population ayant été créés à la même date.
- **Fonction de performance :** (fitness) représente la mesure de performance d'un individu au sein d'une population.

Les AG reposent sur un ensemble de points essentiels qui dominent leur fonctionnement à savoir :

- Un principe de codage des paramètres d'individu. Il existe plusieurs formes de codage variant selon la nature des paramètres à traiter. La qualité du codage conditionne le succès de la recherche.
- Un mécanisme de génération de la population initiale non homogène. Le choix de la population affecte la rapidité de la convergence.
- La formule de la fonction à optimiser doit garantir la prise en charge des facteurs qui ont une influence sur la fitness de l'individu.
- Des opérateurs permettant de diversifier la population au cours des générations et d'explorer des autres solutions possibles.
- Des paramètres de dimensionnement, commençant par la taille de la population ; si ce dernier est trop grand le temps de calcul peut s'avérer très important, tandis que s'il est trop petit l'algorithme peut converger trop rapidement vers un mauvais chromosome. Ensuite, le critère d'arrêt représente un facteur expérimental relatif. Finalement, la probabilité des opérateurs génétiques pendant un cycle influent fortement sur la nouvelle génération ; plus le de croisement élevé plus la population subit des changements importants, plus le taux de mutation est élevé plus de risque d'avoir une solution sous optimal.

1.1.6.7.1 Cycle de fonctionnement

Un AG est un algorithme itératif qui s'exécute dans un cycle de population. Un cycle représente le passage d'une population à la génération suivante. L'évolution génétique d'une population procède en sélectionnant des individus sur la base de leurs pertinences à résoudre le problème relatives à l'espace de solution mesuré par la fonction fitness. Les éléments les plus pertinents ont la chance à se reproduire par l'intervention d'un croisement entre deux chromosomes ou bien une mutation au niveau des gènes. La succession des cycles conduit à

explorer les états possibles et augmente la performance à résoudre le problème posé. Le critère d'arrêt d'algorithme peut être la convergence de l'ensemble des solutions vers le même extremum ou quand le meilleur individu de la population atteint un seuil de performance fixé. Le plus souvent le processus est arrêté au bout d'un nombre d'itérations fixé à priori.

Ce processus peut être schématisé comme suit (figure 1.1) :

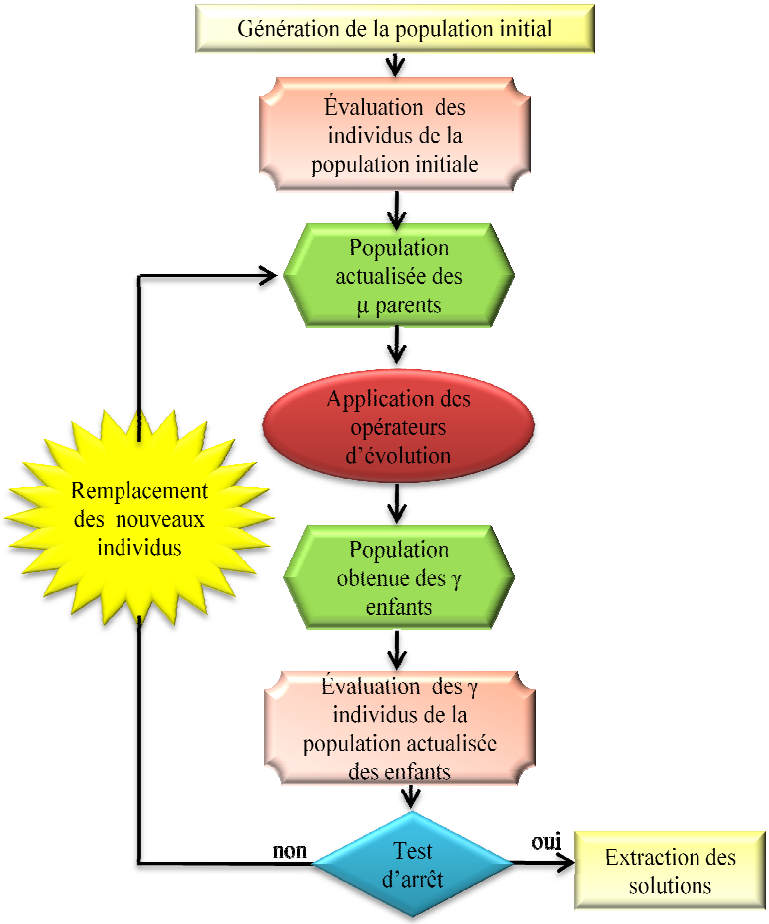

Figure1.1 : Organigramme canonique d'AG

1.1.6.7.2 Codage réel vs codage binaire

Le codage binaire offre l'avantage d'être un mécanisme de représentation et manipulation simple pour un certain nombre de problème. Cependant il existe quelques inconvénients avec ce type de codage :

- Il peut être difficile d'adapter ce codage à certains problèmes. La représentation binaire traditionnelle utilisée pour les algorithmes génétiques crée des difficultés pour les problèmes d'optimisation de grandes dimensions à haute précision numérique. Par exemple, si l'on doit utiliser 100 variables appartenant au

domaine [-500, 500], et si l'on a besoin d'une précision de l'ordre de 10^{-6}, on doit utiliser des chromosomes d'une longueur de l=3000. Cela, en retour, génère un espace de recherche de l'ordre de 10^{1000}. Pour de tels problèmes, les algorithmes génétiques basés sur des représentations binaires ont de faibles performances

- La distance de Hamming entre les codages binaires de deux nombres réels proches peut être assez grande : l'entier 7 correspond à la chaîne binaire 0111 et la chaîne 1000 correspond à l'entier 8, or la distance entre ces deux binaires est de 4. Ce qui créé bien souvent une convergence mais non l'obtention de la valeur optimale.
- Suivant le problème, la résolution de l'algorithme peut être coûteuse en temps.
- Le croisement et la mutation peuvent être inadaptés (pour la création, par exemple, d'individus n'appartenant pas à l'espace de recherche).

Une des améliorations majeures consiste alors à se servir directement de nombres réels. Les résultats donnés par Michalewicz (1992) et Michalewicz et all (1994) ***[114]*** montrent que la représentation réelle aboutit souvent à une meilleure précision et qu'en règle générale le gain en termes de temps de calcul (CPU) est important. Ils en concluent qu'une représentation plus naturelle du problème offre des solutions plus efficaces.

En utilisant le codage réel, l'individu n'est alors plus qu'un nombre réel dans l'espace des valeurs permises: $A = \{a\}, a \in D \subset R$.

1.1.6.7.3 Opérateur génétique

- **sélection des parents**

La sélection est un opérateur clé sur lequel repose en partie la qualité d'un algorithme génétique ***[4, 30]***. Dans cette étape, les chromosomes de la population actuelle sont sélectionnés pour être les parents de la génération suivante. En accord avec la théorie de l'évolution de Darwin, les meilleurs individus doivent survivre et on criée les nouveaux. Il existe plusieurs méthodes pour choisir les individus, tel que la sélection proportionnelle (roulette selection) ***[44]***, la sélection par tournoi ***[29]***, la sélection par rang ***[112]***.

❖ **Croisement**

Le croisement est l'opérateur principal des AG. C'est un opérateur génétique relatif à plusieurs individus parents (souvent deux). Son rôle consiste à combiner les génotypes des individus pour en produire un nouveau. Il fait partie du mécanisme de convergence ***[6]*** de l'AG, qui permet de concentrer la population autour des meilleurs individus. On distingue plusieurs types de croisements possibles. Les plus utilisés sont :

- Croisement à 1 point

Le croisement à un point est l'opérateur de croisement le plus simple et le plus classique. Il consiste à choisir aléatoirement un point de coupure, puis à subdiviser le génotype de chacun des parents en deux parties de part et d'autre de ce point. Les fragments obtenus sont alors échangés pour créer les génotypes des enfants ***[57]***.

- Croisement à multipoints

Le croisement multipoints est une généralisation du croisement à un point. Au lieu de choisir un seul point de coupure, on en sélectionne k, aléatoirement. Dans le croisement multipoints, les points de coupure sont fixés d'avance ***[29]***.

- Croisement uniforme ***[109]***

Ce type de croisement est la généralisation du croisement multipoints. Dans le croisement uniforme, chaque gène d'un enfant est choisi aléatoirement entre les gènes des parents ayant la même position dans le chromosome, avec une probabilité de 0,5 s'il y a deux parents. Le second enfant est construit en prenant les choix complémentaires du premier enfant.

❖ **Mutation**

Les AG utilisent l'opérateur de mutation comme moyen de préserver la diversité de la population ***[73]***. Elle inverse aléatoirement les bits du génotype, avec une faible probabilité.

- ✓ Mutation stochastique (bit flip): Étant la plus employée avec le codage binaire, cette méthode de mutation consiste à inverser indépendamment

chaque bit du chromosome. Un test sur le taux de mutation est effectué pour chacun des bits du chromosome : en cas de succès, le bit testé est alors inversé.

- ✓ Mutation 1 bit : Un bit du chromosome est choisi au hasard. Sa valeur est alors inversée.

1.1.7 Comparaison des techniques d'apprentissage

La comparaison entre les différentes méthodes d'apprentissage est très rare dans la littérature, mais ça n'empêchait pas de trouver quelques études qui illustrent la performance de chacune selon la conception du problème et le mode de fonctionnement des algorithmes. On retient plusieurs paramètres : la simplicité de la méthode, la facilité d'adaptation au problème, la possibilité d'intégrer des connaissances spécifiques, la qualité de la meilleure solution trouvée, et la rapidité en termes de temps de calcul nécessaire pour trouver une solution satisfaisante. ***[12, 54].***

Une étude comparative entre les AG et les RN comme étant deux outils heuristiques est donnée sous forme du tableau suivant (tableau 1.1):

Caractéristiques	Réseaux de neurones	Algorithmes génétiques
Modèle biologique	Système nerveux	Sélection naturelle
Fondement biologique	Apprentissage individuel (cerveau)	Mécanisme d'évolution
Nœud	Neurone	Individu
Dynamique d'état	Fonction d'activation (seuil/poids synaptiques)	Sélection/ crossing-over/ mutation
Traitement	Parallèle	Parallèle implicite⋆
Apprentissage et adaptation	Règles d'apprentissage par des exemples	Évolutif

⋆ propriété qui le permet d'explorer vastes régions de l'espace de recherche tandis qu'il manipule relativement peu de chaînes, cette propriété est démontré par Goldberg et Holland ***[114]*** tel que le nombre d'individu par l'algorithme est au moins proportionnel au cube du nombre d'individu.

Compréhension du problème	Nécessaire	Non nécessaire
Interprétation	Difficile	Facile
Limitation	Exemples d'apprentissage	Optimisation et paramétrisation

Tableau1.1 : Comparaison des caractéristiques des AG et RN (88)

En général, les études de comparaison de performance des AG avec les autres montrent une meilleure performance et une certaine robustesse dans la résolution des problèmes avec leur emploi. Et elle fait la conclusion que les AG offrent la possibilité d'étendre la méthode vers une stratégie d'adaptation et d'apprentissage ***[88]***.

Cependant plusieurs tendances favorisent l'émergence de systèmes hybrides pour renforcer la robustesse du système d'apprentissage ***[36]***.

1.2 Agent et système multi agents

1.2.1 Concept agent

Avant de d'exhiber les caractéristiques de l'apprentissage chez l'agent, il est essentiel de se pencher d'abord sur la notion d'agent. Il existe, à l'heure actuelle, une pléthore de définitions de l'agent. Cela est dû principalement à la relative jeunesse du domaine. Une définition simple donnée par Russel et Norvig stipule qu'« Un agent est tout ce qui peut être vu comme percevant son environnement au travers de capteurs et agissant sur cet environnement au travers d'effecteurs » ***[95]***. Une autre définition consiste qu'« un agent est une entité logicielle ou matérielle, à laquelle est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome, disposant d'une connaissance partielle de ce qui l'entoure (son environnement), et agissant par délégation pour le compte d'une personne ou d'une organisation» ***[89]***.

La définition énoncée par Ferber comporte qu'un agent est une entité logicielle, qui possède des capacités propres de résolution des problèmes et qui est capable d'interagir avec son environnement ***[38]***.

Rahwan***[90]***, Wooldridge et Jennings***[61]***, considèrent un agent comme une entité qui possède un comportement autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

Ces définitions émergent les caractéristiques suivantes :

- L'agent est menu d'une tâche à effectuer (résolution des problèmes) et possède des techniques pour atteindre son objectif.
- L'agent est une entité autonome ; son comportement est déterminé par sa propre expérience exercé dans son propre environnement.
- L'agent dispose d'une connaissance, même partielle, de son environnement courant. Ceci lui permet de prendre les décisions appropriées.
- L'agent est caractérisé par un comportement flexible; il répond d'une manière opportune aux changements qu'il perçoit dans son environnement, il prend l'initiative d'adopter la décision appropriée pour atteindre son but, comme il est capable d'interagir avec les autres agents.

➔ Étant qu'un agent possédant ces caractéristiques, il est habilitant d'apprendre et d'évoluer en fonction de cet apprentissage en modifiant son comportement selon ses expériences passées.

1.2.2 Apprentissage ou rationalité ?

Russell et Norvig ont étendu la définition d'un agent en introduisant le concept de rationalité. ***[94]***

Un Agent rationnel idéal choisi toujours l'action qui l'amène au but final en maximisant la valeur de la mesure de performance. Il agit sur la base de ses connaissances du monde, pour chaque séquence de perceptions La perception active est un aspect important d'un comportement rationnel.

Alors quelle est la différence entre la rationalité et l'apprentissage d'un agent? Pour qu'on puisse répondre à cette question, il faut citer les hypothèses en quoi s'appuie la rationalité. ***[81]***

- ☑ **L'information complète:** Tous les agents sont supposés être entièrement informés du problème.
- ☑ **L'anticipation rationnelle :** l'agent sait au préalable les résultats attendus d'une action et il dispose des critères d'évaluation de leurs actions lui permette de prendre la bonne décision.
- ☑ **La rationalité parfaite:** tous les agents sont supposés être parfaitement capables de déduire leur comportement optimal quelque soit la complexité du problème.

- ☑ **Les espérances communes:** tous les agents agissent selon les mêmes principes de l'information complète et de la rationalité parfaite. Ils savent que les autres savent qu'ils savent.

Le problème posé et que l'on peut aisément remarquer est le rencontre des difficultés dans les cas complexes, justement parce que ces trois suppositions ne sont pas toujours satisfaites.

- × Les agents doivent s'informer du contexte quand le jeu a commencé. Et les problèmes de contexte peuvent eux mêmes ne pas être complètement définis initialement.
- × Les agents ne sont pas toujours assez intelligents ou n'ont pas de capacité d'action pour calculer un optimum.
- × Les agents utilisent différentes approches et ne peuvent compter sur les autres pour dupliquer leur raisonnement.
- × Les conséquences des actions choisies ne sont pas toujours logiques.
- × Ces difficultés mènent à des prévisions qui ne sont pas toujours escomptées.

C'est alors qu'une théorie évolutionnaire a apparu. Son principe est de considérer le problème exactement à l'opposé de la rationalité, avec des agents qui ont initialement une très petite rationalité et connaissances spécialisées de leur domaine d'action. Nos agents sont alors amenés à s'adapter et à apprendre, devenant petit à petit experts dans leur propre domaine. L'un des précurseurs de cette approche est J. Holland ***[58]*** dans ses travaux sur l'inférence et les algorithmes génétiques ***[45]*** en introduisant la notion d'agents artificiels adaptés.

1.2.3 Apprentissage ou adaptation ?

La définition d'un agent s'enrichie à chaque fois qu'on lui attribue une nouvelle caractéristique, (figure 1.2). L'adaptation aussi semble un point fort pour la persistance de l'agent dans son entourage, car l'agent est situé dans un environnement qui change souvent, par conséquent il doit être en mesure à répondre aux dynamiques produits. Cela nécessite une capacité d'adaptation pour qu'il puisse survivre et atteindre ses objectifs.

Le thème de l'apprentissage et de l'adaptation pour les agents est fréquemment évoqué. L'importance de l'apprentissage devrait s'accroître dans les prochaines années tant l'adaptation est primordiale chez les agents ***[37]***.

Quelle relation existe-t-il entre l'apprentissage et l'adaptation ?les propos de G.Weib ***[118]*** ne fait aucune distinction explicite entre les deux concepts, l'adaptation est couverte par l'apprentissage. Il s'agit ici de s'interroger sur la question suivante : comment faire évoluer le comportement des agents de manière à ce qu'ils puissent tirer parti des expériences passées?

Cependant, on peut tirer les points suivants :

- ★ L'adaptation est l'acte résultat face aux changements des situations.
- ★ L'adaptation n'octroie pas des nouvelles connaissances ou comportements. Elle ne sert qu'à modifier les anciens pour les réajuster.

En récapitulation, on peut retenir la définition qui vient d'un agent apprenant, elle correspond le plus mieux au cadre de notre étude.

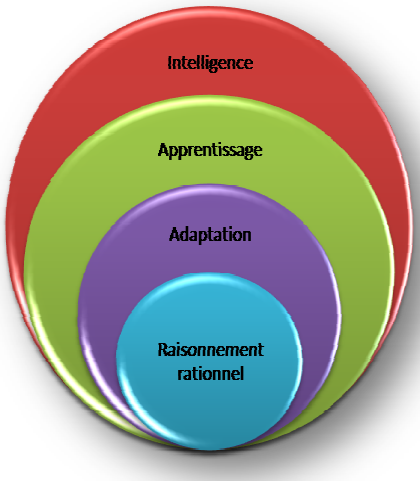

Figure1.2 : Caractéristiques d'un agent

1.2.4 Agent apprenant

Définition :

On dit qu'un agent est entrain d'apprendre si pendant l'exécution de ses tâches il cherche à améliorer ses performances d'exécution et d'acquérir de nouvelles connaissances en fonction de ce qu'il perçoit et la connaissance dont il dispose, par modification de techniques et la prise en compte des changements survenus dans son environnement.

Auprès de ce qui était énoncé auparavant, l'apprentissage est caractérisé par :

- ✓ Une action autonome.
- ✓ Apprendre pour s'évoluer en acquérant des nouvelles connaissances (représentation et comportement).
- ✓ Apprendre pour raisonner avec rationalité.
- ✓ Réciproquement, Apprendre pour assurer son adaptation, et s'adapter par apprentissage.
- ✓ Apprendre pour procurer de l'intelligence.

➪ Si un agent est capable de raisonner face aux situations rencontrées dans son environnement (choisir la meilleure action à exécuter), d'avoir une faculté à s'adapter aux changements produits, et disposé à apprendre des nouvelles procédés pour s'améliorer, alors il est sur le point d'être un agent évolue qualifier d'intelligence (figure 1.3).

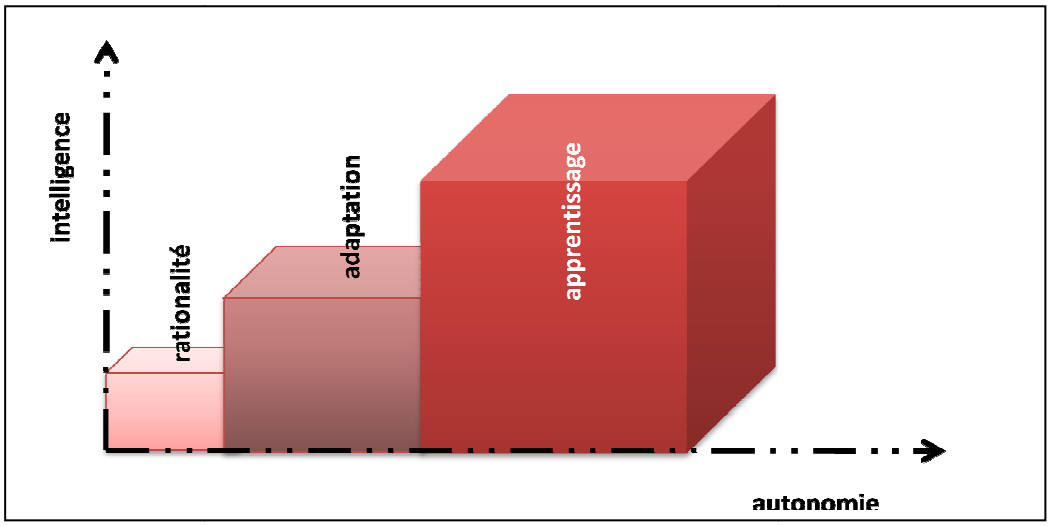

Figure1.3 : Degré d'autonomie et d'intelligence par rapport les caractéristiques d'un agent

Pour que l'agent puisse faire évoluer, il a besoin de deux types d'information. Tout d'abord, il doit avoir des informations sur la manière dont le monde évolue, indépendamment de l'agent, comme il doit avoir ensuite des informations sur la manière dont ses propres actions affectent le monde autour de lui.

1.2.5 Typologie des agents

Il existe diverses classifications qui s'appuient sur les propriétés trouvées en étudiant les diverses définitions pour les agents. Une autre façon de voir un agent réside

dans sa manière d'agitation dans son environnement, à ce stade, on distingue les trois types suivants :

- **Les agents réactifs :** ne font que réagir d'une manière mécanique aux stimuli qu'ils perçoivent. Ils n'ont pas de représentation symbolique de leur monde.
- **Les agents cognitifs :** qui disposent d'une capacité de raisonnement, d'une aptitude à traiter des informations diverses liées au domaine d'application, et des informations liées à la gestion des interactions avec les autres agents et l'environnement.
- **Les agents hybrides :** intègrent l'aspect cognitif et réactif.

1.2.6 Système multi agents

Les chercheurs en intelligence artificielle sont orientés vers une vision distribuée des systèmes pour plusieurs motivations. Plutôt que de considérer un agent unique, compliqué, difficile à maintenir et apparaissant finalement comme une ressource critique, ils ont appliqué le principe "diviser pour régner". Les systèmes multi-agents s'appuient sur le principe suivant : au lieu d'avoir un seul agent en charge de l'intégralité d'un problème, on considère plusieurs agents qui n'ont chacun en charge qu'une partie de ce problème. La solution au problème initial est alors obtenue à travers de l'ensemble des comportements individuels et des interactions, c'est à dire par une résolution collective***[33]***.

Définition :

Dans Ferber 1999 ***[38]***, un système multi-agent est défini de la façon qui suit: On appelle système multi-agent (ou SMA), un système composé des éléments:

1. Un environnement E, c'est-à-dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets O. Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
3. Un ensemble A d'agents, qui sont des objets particuliers ($A \subset O$), lesquels représentent les entités actives du système.
4. Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
5. Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.

6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

1.2.7 L'apprentissage chez les agents

Pour la plupart des tâches que doit accomplir les agents et même si l'environnement paraît plus ou moins stable, il est extrêmement difficile et même impossible à priori de déterminer correctement le comportement et les activités concrètes de ce système au moment de sa conception. Cela nécessite de connaître à priori la nature de l'environnement émerge et le genre d'agents qui pourront s'adapter à ce nouvel environnement. Ce genre de problèmes qui résultent de la complexité des systèmes multi-agents peuvent être évités ou du moins réduits en octroyant aux agents la possibilité de s'adapter et d'apprendre avec la possibilité d'améliorer leur propre performance ou/et celle du système dans son ensemble **[118]**.

1.2.7.1 Apprentissage mono agent

L'idée derrière l'apprentissage, c'est que les perceptions de l'agent ne devraient pas être utilisées seulement pour choisir des actions, elles devraient être aussi utilisées pour améliorer l'habilité de l'agent à agir dans le futur. L'apprentissage, pour un agent, est très important car c'est ce qui lui permet d'évoluer, de s'adapter et de s'améliorer ***[95]***.

Voir l'apprentissage chez les êtres humains, plus que les réflexes innés, comme retirer la main lorsque l'on se brûle, ils peuvent apprendre de nouveaux réflexes plus compliqués comme la conduite automobile. Au début, la conduite est très difficile, car on doit penser à toutes les actions que l'on fait, mais plus on se pratique, moins on réfléchit et plus la conduite devient un réflexe.

Dans la forme la plus simple, on peut appliquer la même chose. C'est-à-dire, lorsqu'un agent fait face à une situation pour la première fois, il doit délibérer plus longtemps pour choisir ses actions. Mais, avec un module d'apprentissage, plus l'agent effectue des tâches similaires, plus il devient rapide. Son comportement passe graduellement d'un état délibératif, à un état réactif. L'agent a donc appris à exécuter une tâche. D'un point de vue plus technique, on peut dire que l'agent a, en quelque sorte, compilé son raisonnement dans une certaine forme d'ensemble de règles qui lui permettent de diminuer son temps de réflexion. Ce n'est qu'une façon dont les agents peuvent apprendre, il en existe plusieurs autres. En fait, on considère que toute technique qui permet à un agent d'améliorer son efficacité est une technique d'apprentissage.

L'apprentissage individuel consiste qu'un agent peut apprendre de façon solitaire et indépendante des autres agents ***[119]***. Il n'est pris en considération que lorsque tous les éléments du processus d'apprentissage sont exécutés par le même agent et n'incluse que ses propres expériences ***[118]***. À ce stade plusieurs travaux sont considérés intéressants, ils implémentent différentes techniques d'apprentissage dans divers domaines d'application. Entre autre nous citons Baird & Moore 1998 ***[5]***, Kearns & Singh 1998***[65]***, Satinder Singh 2000 ***[105]***, Bruno Scherrer 2003***[98]***, Toni Conde 2005***[20]***, Nicolas Gomond et Jean-Marc Salotti 2006***[46]***.

1.2.7.2 Apprentissage multi agents

Il l'étend bien au delà dans la mesure où les activités d'apprentissage d'un agent peuvent être influencées considérablement par les autres agents et que plusieurs agents peuvent apprendre de manière distribuée et interactive comme une seule entité cohérente ***[60, 103]***.

Les SMA évoluent généralement dans des environnements complexes (c'est-à-dire larges, ouverts, dynamiques et non prévisibles) ***[104]***. Pour de tels environnements, c'est très difficile et même quelquefois impossible de définir correctement et complètement les systèmes à priori. Ceci exigerait de connaître à l'avance toutes les conditions environnementales qui vont survenir dans le futur, quels agents seront disponibles à ce moment et comment les agents disponibles devront réagir et interagir en réponse à ces conditions. Une manière de gérer ces difficultés est de donner à chaque agent l'habilité d'améliorer ses propres performances, ainsi que celles du groupe auquel il appartient.

Dans un environnement multi agent, les agents peuvent apprendre grâce aux autres. Par exemple, un agent A, qui voudrait savoir comment se rendre à un certain endroit, pourrait demander à un autre agent B s'il connaît un bon chemin. Si B connaît un bon chemin, il peut le transmettre à A, permettant ainsi à l'agent A d'apprendre un bon chemin grâce à l'agent B ***[41]***.

D'un autre côté, les agents peuvent aussi apprendre à propos des autres. Par exemple, un agent peut regarder un autre agent agir dans certaines situations et, à l'aide de ces

informations, il pourrait construire un modèle du comportement de l'autre agent. Ce modèle pourrait lui servir pour prédire les actions de l'autre agent dans le futur. Cette information pourrait l'aider à mieux se coordonner ou à mieux collaborer avec l'autre agent ***[42***, ***79]***.

Une autre facette importante est l'apprentissage des interactions ***[8]*** (coordination, collaboration, communication, etc.) entre les agents pour échanger leurs expériences, l'hypothèse sous-jacente est que pourquoi passer un temps assez important pour apprendre une chose qu'un autre dans mon groupe la connaît. Ainsi que l'apprentissage collectif augmente la performance du système ***[16]***.

Plusieurs modèles ont été proposés pour représenter l'apprentissage multi agent, Tan 1993***[110]***, Hu & Wellman 1998***[60]***, Brafman & Tennenholtz 2001***[13]***, Conitzer et Sandholm 2003***[21]***, Alain Dutech 2006***[34]***.

Pendant notre étude, nous somme intéressés par l'apprentissage mono agent.

1.3 Apprentissage par AG

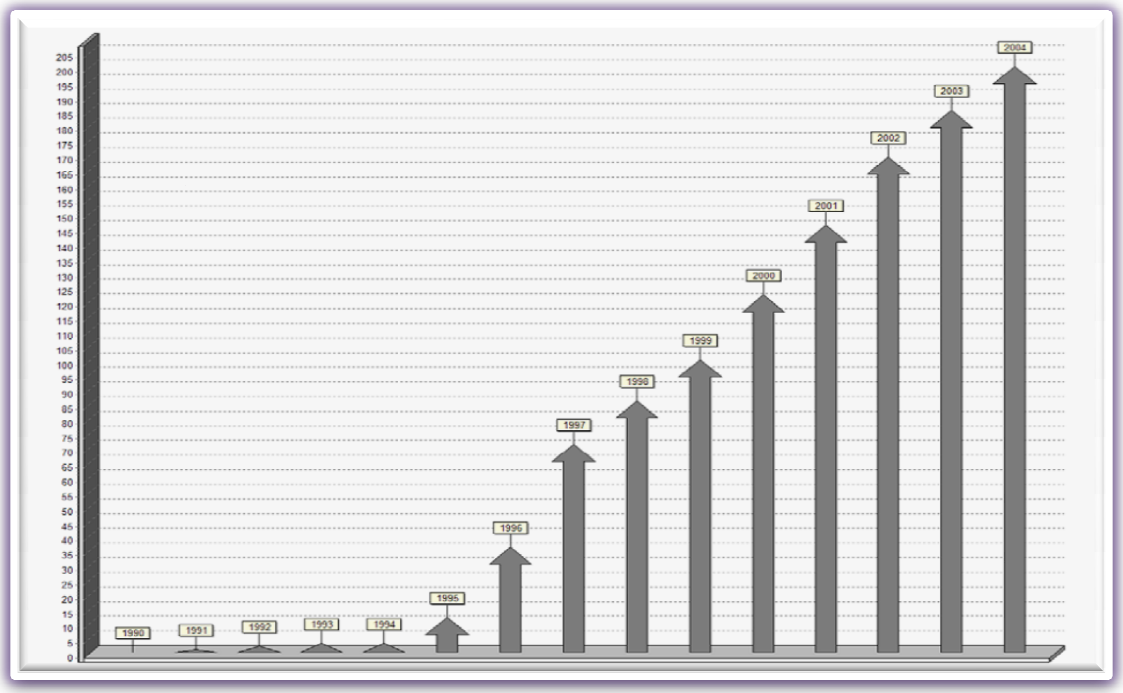

Figure1.4 : Évolution cumulée du nombre de publications utilisant les AG [114]

On constate un nombre croissant de travaux qui implémentent les AGs, la figure 1.4 résume les publications énoncées depuis 1990 jusqu'à 2004 avec leurs énumération pendant

chaque période. On observe une croissance très forte de l'utilisation des AG depuis 1994, il y a 200 articles publiés avant le premier janvier 2004.

Les AG ont été appliqués avec succès dans des problèmes d'optimisation et d'apprentissage automatique ***[18, 39, 64, 76, 96]***. Ceux-ci sont particulièrement efficaces pour parcourir des espaces de recherche à la fois vastes et complexes. Ils ne nécessitent pas notamment de connaissances spécifiques du domaine pour converger vers une bonne solution, mais au contraire mettent en œuvre un cadre général permettant d'évaluer et de recombiner des solutions partielles.

De Jong , 1988 ***[29]*** rapporte que les AG sont plus performants que d'autres techniques d'apprentissage pour la conception de systèmes d'apprentissage. Cette performance est également comparée dans les travaux de Grenfenstte 1993 ***[48],*** qui présente une rétrospective sur l'application des algorithmes génétiques dans le domaine des algorithmes d'apprentissage. Plusieurs auteurs ont également employé les AG spécifiquement pour l'acquisition de connaissances dans les systèmes de classificateurs Knight et Sen 1995***[67]***, Fàbrega et Guiu 1999 ***[35]***. Holmes 2002 ***[59]*** présente un bon exposé sur le sujet dans lequel de nouveaux modèles sont discutés et plusieurs applications intéressantes sont présentées. Des travaux de recherches concernant les systèmes multi-agents proposent aussi d'employer les AG comme système d'apprentissage à l'intérieur des agents intelligents. Cette approche permet aux agents d'être complètement autonomes d'un point de vue génétique ***[18]***. Cette autonomie génétique permet un apprentissage de haut niveau, par exemple, de la détermination du meilleur jeu d'heuristiques au cœur même des agents.

Précisément, les recherches récentes en économie pour adopter les AG comme outil d'apprentissage sont encourageantes. Nous venons présenter les plus récents et proches de notre travail.

Schulenburg et Ross 2000 ***[99]***, établent la performance des agents artificiels face aux données provenant d'un marché financier réel. Les agents sont d'abord entraînés sur des données des neufs ans et ensuite ils utilisent les règles qu'ils ont développées pendant cette période pour gérer leurs transactions pendant la dixième année. Les simulations montrent que les agents artificiels développent une large diversité de stratégies innovantes qui ont une performance supérieure à celle des stratégies de base de type « buy and hold ».

Vriend 2000 ***[117]***, propose l'utilisation des AG pour l'étude des dynamiques d'apprentissage et il obtient la convergence vers la solution de Cournot avec l'apprentissage individuel.

Vallée 2000 ***[115]***, utilise les AG pour étudier les ajustements dans un modèle de jeu répété, consacré à la crédibilité de la politique monétaire du gouvernement. Après l'annonce du gouvernement, l'évolution de l'AG correspond à l'apprentissage, par les agents, de la fonction de réaction du gouvernement en fonction de l'évolution de l'inflation réalisée. Cet article teste différentes structures possibles pour un AG avec codage réel et cherche à mesurer l'information générée par chaque structure et son impact sur l'évolution du jeu. L'intérêt est qu'on doit porter à la structure de l'AG retenu et la nature de l'information générée et traité par le AG.

Yildizoglu 2001 ***[120]***, étudie la pertinence des AG pour la modélisation de l'apprentissage individuel dans les stratégies de R&D[⋆] des firmes. Il développe un modèle de concurrence entre deux types de firmes les NWFirms qui utilisent une règle fixe pour arbitrer entre l'investissement en R&D et celui en capital physique, et les GenFirms qui utilisent un AG individuel pour ajuster cet arbitrage à l'évolution de leur industrie. Les principaux résultats de ce modèle montrent que cet apprentissage individuel est une source d'avantage concurrentiel pour les GenFirms qui finissent par dominer l'industrie.

Jasmina Arifovic, Michael K. Maschek 2006***[1]***, éprouvent l'apprentissage individuel des firmes pour mettre à jour leurs décisions sur la quantité de productions. Ils ont illustré la convergence du modèle. Leur conclusion était que cette convergence est due grâce aux deux points : la manière d'évaluation des performances de la production des règles, plus la fonction du coût.

1.4 Apprentissage par renforcement (AR)

L'AR est une approche computationnelle pour apprendre et automatiser le « goal-direct learning » (apprentissage dirigé but) et la prise de décision. Il se distingue des autres approches computationnelles par le fait que l'apprentissage se fait de façon individuelle à partir des interactions directes de l'environnement sans avoir recours à des exemples de supervision ou à des modèles spécifiques de l'environnement. L'apprentissage par

[⋆] Une stratégie économique « recherche et développement », fonction de l'entreprise qui se charge des nouveaux produits, de la conception à la réalisation ***[120]***.

renforcement utilise un framework formel définissant les interactions entre un agent apprenant et son environnement, en termes d'états, d'actions et de récompenses (rewards). L'agent apprenant doit pouvoir prendre connaissance sur l'état de son environnement et être capable d'agir en conséquence pour affecter son environnement. Il exploite ses connaissances antérieures pour obtenir des récompenses, mais il doit également effectuer une opération d'exploration afin de faire la meilleure sélection d'actions dans le futur, c'est-à-dire qu'il doit essayer une variété d'actions et choisir progressivement celles qui paraissent les plus performantes ***[8]***.

Tesauro 2002 ***[111]***, a utilisé l'AR pour modéliser des décisions économiques consistant à fixer les prix dans un marché compétitif. Cette décision est prise dans un environnement où tous les agents sont adaptatifs, ce qui en fait un environnement évolutif et dépendant de l'historique. Cet algorithme a été utilisé dans un jeu de taille réduite, limité à deux agents, pour que l'espace des états ne soit pas trop grand et puisse être représenté par les tables de Q-functions.

Tandis que Kutchinski 2003***[69]***, a développé un système d'AR pour déterminer les stratégies des vendeurs permettant de fixer les prix sur un marché. L'utilisation du Q-learning⋆ a donné des meilleurs résultats que ceux obtenus par l'utilisation de règles simples pour de tels systèmes.

1.5 Conclusion

Pour que la machine soit un vrai collaborateur avec son utilisateur, en reconnaissant bien son environnement et en prenant des décisions correctes, avec une intervention extérieure limitée, l'apprentissage artificiel offre la capacité de faire évoluer ses connaissances et progresser ses performances, afin de maîtriser en cours du temps la tâche à réaliser.

Le processus d'apprentissage possède des caractéristiques concernant les données à utiliser, le temps à consommer et le rôle à jouer par l'enseignant. Ainsi que certaines conditions doivent être vérifiées au sein du système apprenant. Car l'apprentissage est une notion multi concepts, diverses classifications ont été proposées dans la littérature en tenant compte chaque fois un critère spécifique. L'apprentissage est un domaine multidisciplinaire qui fait l'objet de nombreux domaines, par conséquent différents outils et techniques ont été développés pour répondre aux exigences des problèmes en discussion.

⋆ Le *Q-learning* est une technique qui se place dans le cadre de l'apprentissage par renforcement et qui produit une matrice *Q* dans laquelle chaque élément *Q(s,a)* mesure l'intérêt d'effectuer l'action *a* lorsque l'on se trouve dans l'état *s[8]*.

L’agent comme entité logicielle ou matérielle a besoin, dans la majorité des cas, d’un processus d’apprentissage, pour qu’il devienne plus puissant et performant à résoudre les problèmes dont il rencontre dans son environnement et d’agir efficacement.

Selon les recherches en apprentissage, plusieurs méthodes ont été développées. Les AG ont prouvé leurs succès, ainsi que l’AR s’avère un outil qui offre l’agent la possibilité de maximiser ses récompenses.

Nous avons donc besoin de concevoir une architecture d’un agent apte à apprendre, en développant l’AG utilisé comme un processus d’apprentissage avec l’intégration d’AR. Ce sont les deux points dont le chapitre suivant fait l’objet.

Chapitre 2

Conception du modèle d'un agent apprenant basé algorithme génétique

L'une des problématiques liées à l'intelligence artificielle est le fait de mener un agent par une caractéristique qui le rend capable d'apprendre automatiquement depuis son environnement où il existe. L'objectif est de le permettre d'évoluer au cours du temps, en améliorant ses performances et augmentant ses connaissances.

Cette partie est consacrée à la proposition d'une architecture d'agent apprenant, en incluant un algorithme génétique pour formaliser le problème d'apprentissage, incorporé avec un mécanisme d'apprentissage par renforcement.

Nous spécifiant les éléments intervenant dans l'architecture proposée. Ainsi que les étapes à inclure dans le processus génétique, précisément les opérateurs adoptés. L'utilisation d'un apprentissage par renforcement exige la définition d'une fonction d'évaluation des performances qui est détaillé à ce niveau. Finalement les protocoles globales à suivre sont décrits chacun séparément.

2.1 Nécessité d'une approche d'apprentissage par algorithme génétique

L'apprentissage est une aptitude nécessaire pour amener un système à évoluer de manière efficace tenant compte les circonstances de son environnement.

Le contexte d'utilisation des agents dans un milieu dynamique, l'apprentissage semble être une faculté indispensable, tant pour l'amélioration de la performance d'agent, que pour lui permettre à disposer même si les connaissances sont modiques.

C'est pourquoi, différents outils d'apprentissage ont été développés. Après le succès réalisé par les algorithmes génétiques à résoudre des problèmes d'optimisation, ils sont aujourd'hui suggérés d'être utilisés comme un mécanisme d'apprentissage. Cette tendance est également liée à leur caractère évolutif. Ainsi que leurs capacités à s'adapter à n'importe quel espace de recherche bien que les connaissances initiales sont restreintes et les mouvements d'environnement sont imprévus. L'utilisation des AG offre la possibilité d'obtenir rapidement une solution de bonne qualité et ce assez facilement comparativement à d'autres techniques.

L'apprentissage ne vise pas seulement à élaborer la bonne décision, mais aussi de prendre en compte l'effet correspondant à cette décision pour s'améliorer. L'agent ne garantie l'efficacité de son processus d'apprentissage qu'après l'évaluation de ses actions produites.

En conséquence, on est appelé à intégrer un mécanisme d'apprentissage par renforcement afin de permettre à l'agent de garder une traçabilité de ses expériences passées et de mesurer sa qualité d'agitation.

L'hybridation entre les algorithmes génétiques et l'apprentissage par renforcement engendre un outil apte à explorer une stratégie adéquate en rectifiant les décisions par l'utilisation des algorithmes génétiques suite aux valeurs mesuré des performances interprétés comme des renforcements (récompense ou pénalité).

2.2 Architecture proposée

L'architecture proposée d'un agent apprenant est évoqué de celle définit par Russell et Norvig ***[94]***. Elle peut être raffinée en différents niveaux. Une vue de haut niveau d'abstraction de notre modèle montre l'agent composé de trois modules décris comme suit (figure2.1).

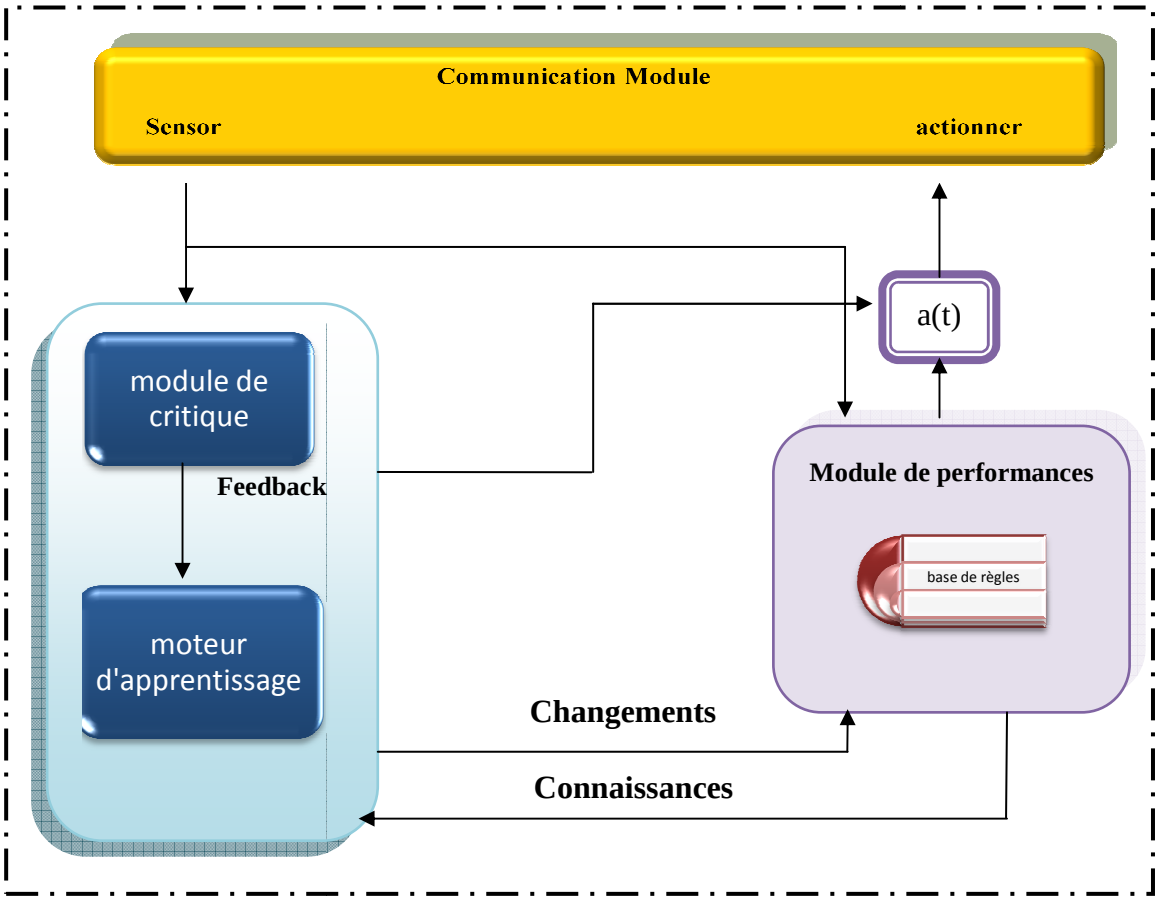

Figure 2.1 : Architecture globale d'un agent apprenant

2.2.1 Module de communication : pour résoudre des problèmes, il faut doter l'agent de connaissances nécessaires à la résolution. Ces connaissances seront obtenues par le biais d'une interface chargé à transférer les évènements apparents dans son environnement déclenchés par ce dernier ou bien suite à une action propre de l'agent. D'autre part, ce module consiste à interpréter les décisions prises de l'agent sous forme d'actions effectuées dans le monde extérieur où il se situe.

2.2.2 Module de performance : son rôle demeure toujours de choisir des actions à effectuer, en se basant sur les perceptions en entrée, ainsi que les informations retournées par le module d'apprentissage. Son choix est effectué sur quel règle à appliquer. Il inclut une base de règle contenant celles appris suite aux cycles d'apprentissage.

2.2.3 Module d'apprentissage : responsable des améliorations du comportement via l'exploration des nouvelles règles. Il reçoit des connaissances du module de performance et un feedback sur la qualité des actions faites par l'agent. Il décide à

partir de ces données de la manière dont le module de performance devrait être modifié afin que l'agent fasse mieux dans le futur. De même, il est constitue de deux unités : une unité de critique et un noyau d'apprentissage (figure2.2). Une description plus raffiné est la suivante.

2.2.3.1 Module de critique: il s'occupe à mesurer les performances de l'agent au file de temps en incluant ses expériences passées et indique au module « moteur d'apprentissage » à quel point agit il bien. Ce module est composé de deux unités (figure2.3).

a. **Une unité de mémorisation:** les renforcements obtenus seront stockés dans une mémoire sous forme d'une matrice permettant d'enregistrer pour chaque état x(t) sa valeur de renforcement correspondante. La mémoire offre à l'agent de s'évoluer en tenant compte les critiques reçus dans ses expériences passées.

b. **Une unité d'évaluation :** fait appel à un processus d'apprentissage par renforcement. Il sert à associer à un instant t, une valeur de renforcement d'une action a(t), correspondante à un état perçu x(t). Un renforcement estimé positif signifie que l'agent a bien agit et son processus d'apprentissage fonctionne d'une bonne manière. En revanche, le renforcement négatif indique un échec, et par conséquent la stratégie adaptée au niveau d'apprentissage doit être modifiée.

✍ **Calcul des renforcements**

➢ **Quelques définitions de base :**

Définition1 : *Une action a(t) d'un agent à l'instant t, est choisit en un premier temps par le processus d'apprentissage génétique, et apprécier en un second temps par le processus d'évaluation.*

Définition2 : *Un rendement r(t) d'une action a(t), est égale à la variation de performance de l'agent entre t-1 et t.*

$$r(t) = P(t) - P(t-1) \qquad (2.1)$$

Équation 2 .1 : Calcul de rendement d'une action

P : performance.

Définition3 : *Le paramètre de renforcement G est un assignement de crédit qui est affecté à chaque retour de l'environnement (récompense ou pénalité) selon les rendements de l'action d'agent réalisée dans l'environnement.*

$$G(t) = \lambda \qquad si \qquad r(t) > 0$$

$$G(t) = -\lambda \qquad si \qquad r(t) < 0$$

$$G(t) = 0 \qquad sinon$$

Définition4 : *Un taux des cumuls positifs des assignements de crédit G^+ est le pourcentage de récompenses positives gagnées suite aux actions exécutées dans le passé par rapport au nombre total des assignements des crédits obtenus.*

$$G^+ = \frac{\sum_{i=1}^{t} G_i^+}{\sum_{i=1}^{t} G_i} \times 100 \qquad (2.2)$$

Équation 2 .2 : Calcul de récompenses positives

Définition5 : *Un taux des cumuls négatif des assignements de crédit G^- est le pourcentage de récompenses négatives gagnées suite aux actions exécutées dans le passé par rapport le nombre total des assignements des crédits obtenus.*

$$G^- = \frac{\sum_{i=1}^{t} G_i^-}{\sum_{i=1}^{t} G_i} \times 100 \qquad (2.3)$$

Équation 2 .3 : Calcul de récompenses négatives

Définition6 : *La mesure de performance M d'un agent (processus d'apprentissage) à l'instant t, est définie par deux valeurs, le taux des assignements de crédit positifs et négatifs, G^+et G^- successivement, cumulé jusqu'à l'instant t.*

$$M = +G^+ \qquad si \qquad G^+ > G^-$$

$$M = -G^- \qquad si \qquad G^- > G^+$$

- **Le protocole d'apprentissage :** à chaque temps t, l'agent suit le protocole suivant pour évaluer la performance d'apprentissage :

1. L'agent est dans un état x(t), il reçoit les nouvelles données de l'environnement.
2. Il réactualise ses données et génère son rendement r(t) et son assignement de crédit G correspondant.
3. Enregistrement r(t) et G(t).
4. Calcul G^+ et G^-.
5. Génération de mesure de performance M(t).
6. Si M(t) < 0 alors envoyer un signal de changement pour le processus d'apprentissage. Sinon rien n'est changé.

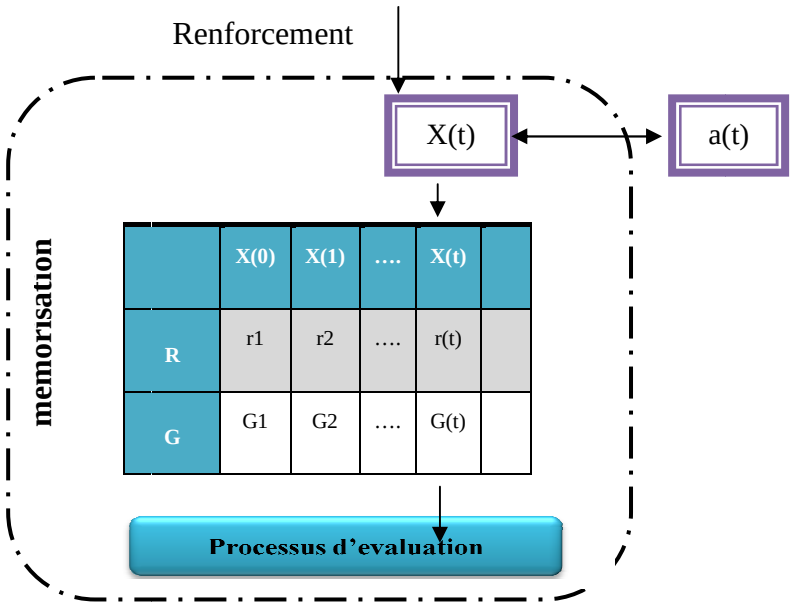

Figure 2.2 : Architecture d'unité de critique

2.2.3.2 Moteur d'apprentissage : il est constitue de deux parties suivantes :

a) **Le générateur de problèmes :** suggère des problèmes exploratoires afin de permettre au système de découvrir des cas qu'il n'aurait pas rationnellement choisis.

b) **processus d'apprentissage :** le principe de fonctionnement se déroule selon les étapes d'un algorithme génétique.

   i. Il repose sur un codage des problèmes et de leurs solutions sous la forme de chaîne d'éléments de base. Les chaînes peuvent être rompus entre chaque élément de base, à l'image des chromosomes qui eux constituent de véritables listes de caractéristiques d'un individu.

ii. On génère tout d'abord une population de solutions potentielles à un problème donné, sous la forme de telles chaînes, puis on sélectionne, au moyen d'une mesure d'ajustement, les éléments de la population qui satisfont au mieux les contraintes de la solution recherchée.
iii. Des opérateurs génétiques sont ensuite appliqués à cette population de manière à obtenir une nouvelle population, possédant dans son ensemble de meilleures solutions que la précédente génération.
iv. On réitère le processus sur l'ième génération, jusqu'à l'obtention d'une génération de solutions qui satisfait les critères de qualité imposée et qui est beaucoup plus adapté au traitement du problème en question. Il suffit alors de choisir la meilleure solution.

Les AG mettent en œuvre de mécanisme assez simple pour l'évolution de la solution. La manipulation des chaînes sont faites en utilisant des opérateurs basiques : le croisement et la mutation.

Le processus d'apprentissage est efficient s'il apporte des améliorations efficaces sur les décisions d'un agent, l'un des points forts de son efficacité est qu'il est apte à répondre, au bon moment, dans tel telle situation.

Cette aptitude devenus de plus en plus une caractéristique exigée, si l'environnement subit des changements rapides. Par conséquent, la vitesse de convergence d'algorithme génétique vers la solution optimale doit être maintenue vis-à-vis la diversité de la génération qu'elle appartient. Cette contrainte est assurée par l'utilisation adéquate des opérateurs de croisement et mutation adaptés au problème à résoudre.

Notre démarche consiste à implémenter deux variantes de croisement et de mutation afin d'augmenter la performance d'un algorithme génétique.

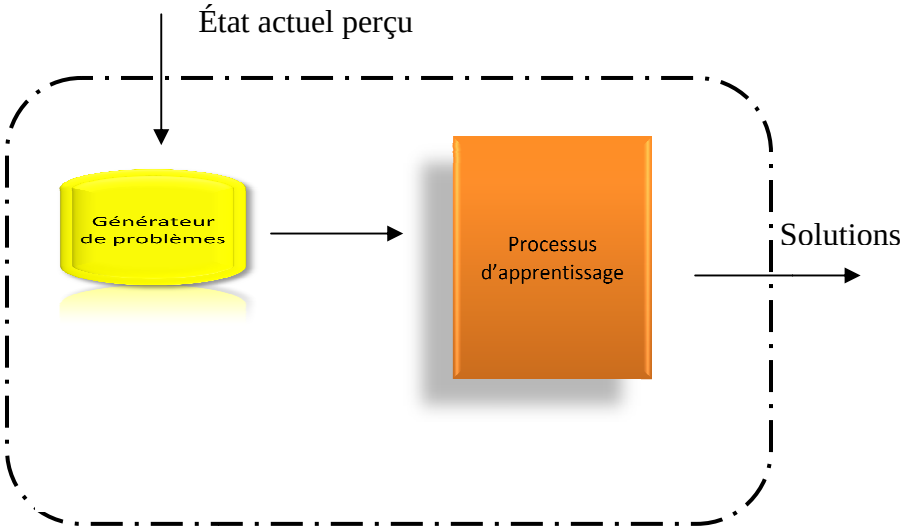

Figure 2.3 : Architecture du moteur d'apprentissage

2.3 Croisement adapté

Cet opérateur a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux individus parents et génèrent deux nouveaux enfants.

Le croisement traditionnel consiste à choisir aléatoirement une position Xp, sur deux chaînes parents composées de M gènes, portant des valeurs de fitness aléatoires. Ensuite les deux sous chaînes terminales de chacune sont échangées ce qui produit deux chaînes filles, figure2.5. Le point de croisement peut être augmenté à 2 ou plus qu'un seul emplacement selon le problème à traiter.

Une telle méthode risque de favoriser la dominance d'un ensemble de caractéristiques dans une génération ce qui empêche la dissemblance et mène le système à tomber dans un minima local.

Pour éviter le rassemblement des individus autour d'un dominant, nous suggérons d'utiliser une alternative de croisement proposée par Angel Kuri.M ***[68]***, qui consiste à faire croiser non plus les deux meilleurs parents, mais plutôt de recombiner deux chromosomes l'un est le meilleur, l'autre est le plus mauvais. La recombinaison s'accomplie par un croisement en anneau (ring crossover) selon le procédé qui suit :

- Classer les (n) individus de la population selon leurs fitness, de tel façon que le premier rang (1) réfère au meilleur individu et le $n^{ième}$ rang réfère au plus mauvais.

- Former un ensemble $\eta = n/2$ de couples ; le premier couple constitué du premier (1) et le dernier(n) des individus ordonnés ; le suivant correspondant aux individus 2 et n-1, etc. D'une manière générale le couple (i) est constitué des individus i et n-i+1. Dans chaque couple le meilleur individu désigne la source, l'autre désigne la cible.
- Avec une probabilité de croisement Pc, pour chaque couple appliquer :
  - Sélection aléatoire de 2 positions de croisement Xp ; la première correspondant à la source, la deuxième correspondant à la cible.
  - Un segment de longueur Ls = O(P)/2 est sélectionné à la cible comme à la source. Les segments sont considérés comme anneau (ring) par le rapprochement de l'extrémité droite du segment vers l'extrémité gauche
- Finalement, un croisement est effectué entre la source et la cible.

2.3.1 Exemple illustratif

Le croisement adapté est illustré à travers un exemple représenté par les figures 2.4 et 2.6. La longueur de la chaîne représentant l'individu s = O(P) = 6. Les deux individus sont : $U_1(t)$ = 100101, $U_n(t)$ = 101010. Le segment à croiser de longueur Ls = O(P)/2 = 3.Le point de croisement aléatoire X_p = 4. Après un croisement anneau (ring crossover), les deux chaînes obtenus sont : $U_1(t+1)$ = 100101, $U_n(t+2)$ = 101010.

Figure 2.4 : Les couples à croiser

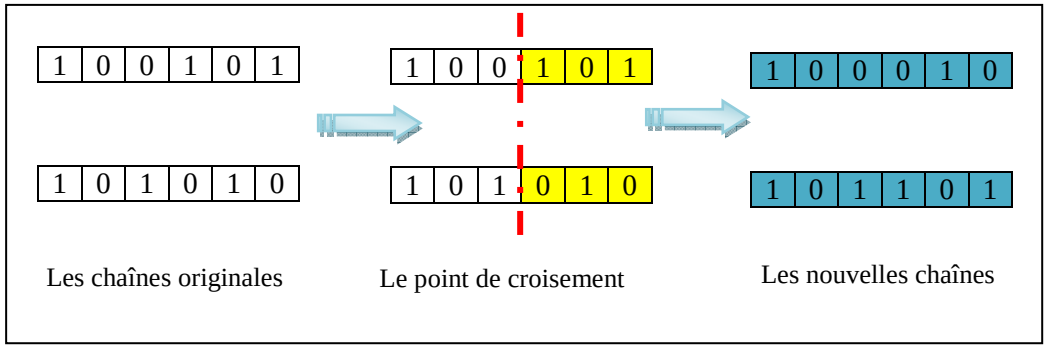

Figure 2.5 : Le croisement linéaire

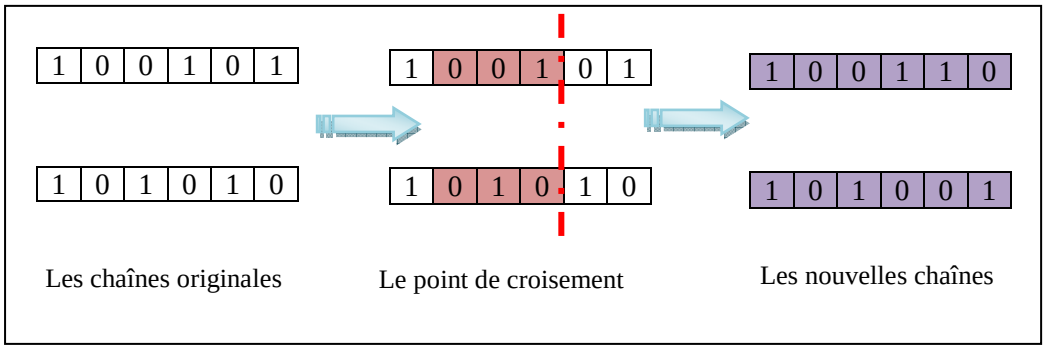

Figure 2.6 : Le croisement en anneau (ring crossover)

2.4 Mutation adaptée

L'opérateur de mutation apporte aux algorithmes génétiques la propriété d'ergodicité de parcours d'espace de solutions. Cette propriété indique que l'AG sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de résolution.

La mutation traditionnelle consiste généralement, à tirer aléatoirement un gène dans le chromosome et le remplacer par une valeur aléatoire. Ce changement est une étape critique ; il peut rapprocher l'individu vers la solution optimale, comme il peut l'écarter largement et la reprise peut être difficile par la suite.

Une des tendances sert à ajuster le taux de mutation, tandis que le taux reste une mesure probable et indéterminée, varie selon le problème rencontré.

Vu l'influence de cet opérateur sur la convergence d'un AG vers la solution optimale, et ainsi d'améliorer le rendement de processus d'apprentissage. Il est spontané de penser à implémenter la mutation d'une manière permettant au système d'éviter le problème de convergence prématuré ou d'écarter de la bonne solution.

Répliquant, à cette exigence, nous proposons d'incorporer dans le cycle génétique, un opérateur de mutation inspiré du travaux de Sung Hoon Jung ***[62]***, nommé

« mutation sélective » dont le but est d'augmenter la puissance exploratoire de la mutation et par conséquent réduit le temps nécessaire pour l'apprentissage.

Sung.Jung a constaté que la mutation est significative s'elle est effectuée dans la partie chromosomique adéquate. En se basant sur cette proposition, la mutation sélective consiste à remplacer un gène selon la fitness de son individu y appartient ; si un individu possède une basse fitness, il est plus loin d'être une solution optimale, alors le changement aura lieu dans la partie significative du chromosome pour qu'il fait un saut rapide. Par contre, si l'individu a une fitness élevée, il est plus probable d'être proche de la solution optimale. Et ainsi le point à remplacer est inclus dans la partie moins signifiante pour qu'il reste autour de la solution optimale.

D'autre part, le codage binaire des chromosomes est favorable pour renforcer l'utilisation de cet opérateur, car la représentation binaire des chaînes de chaque individu désigne que la position du bit la plus élevée implique que c'est lui-même qui a le poids le plus fort. L'opérateur adopté se déroule à travers les étapes qui suivent.

Mutation Sélective

n : La taille de la population.

N : Le nombre de partition.

s : La longueur de la chaîne chromosomique.

r_i : Le rang de l'ième individu.

$P^s_{1,\ldots,N}$: Une partie de chaîne d'individu.

$P^r_{1,\ldots,N}$: Une partie du rang.

1.Trier les individus de la population selon leurs fitness et leur rang.

2.Diviser la chaîne chromosomique de chacun en N parties, P^s.

3.Diviser le rang en N parties, P^r.

Pour i = 1 à n faire

Si $r_i \in P^r_j$ alors

Muter un point sélectionné aléatoirement, appartient à la partie,P^s_{N-j+1}.

Fin si

Fin pour.

Algorithme 2.1 : Mutation sélective

D'après l'algorithme proposé, les individus de la population sont classés suivant leurs fitness, et un rang correspondant est affecté à chacun. Ensuite les chaînes des chromosomes sont segmentées en N fragments. Le bit à modifier sera tiré aléatoirement du segment P^s_{N-j+1} de la chaîne, si le rang de chaque individu se situe dans la partie P^r_j du rang.

2.4.1 Exemple illustratif

Pour développer l'utilisation de la mutation sélective, l'exemple suivant montre son procédé. Supposant que la taille de la population n = 20, la longueur de la chaîne d'individu s= 16 et le nombre de partition N=2, ensuite les chaînes sont divisées en deux parties $P^s_{1,2}$, de même, le rang est diviser en deux $P^r_{1,2}$. Si le rang d'un individu est 3, alors il appartient à la première partie P^r_1. Par conséquent un bit aléatoire inclut dans la deuxième partie P^s_2 de la chaîne est changé. Cependant, si le rang de l'individu se trouve dans la deuxième partie P^r_2, alors un bit aléatoires dans la première partie P^s_1 est modifié.

2.5 L'algorithme génétique global adapté

L'algorithme génétique adapté se déroule à travers des étapes décrites comme suit précédé par une définition de représentation génétique du problème à traiter.

Step1 : *créer une population initiale des individus P(0).*

Step2 : *évaluer la fitness de chaque individu.*

Step3 : *sélection d'un opérateur génétique*

- *Avec une probabilité de croisement Pc.*
- *Si le feedback reçu est négatif alors appliquer le croisement adapté.*
- *Sinon, appliquer le croisement classique.*
- *Avec une probabilité de mutation Pm.*
- *Appliquer la mutation classique.*
- *Puis appliquer la mutation sélective.*
- *Reproduction du meilleur individu.*

Step4 : *ajouter le nouvel individu à la population, avoir P(t+1).*

Step5 : *si critère non satisfait, aller au* ***step 2****.*

Algorithme 2.2 : AG adapté

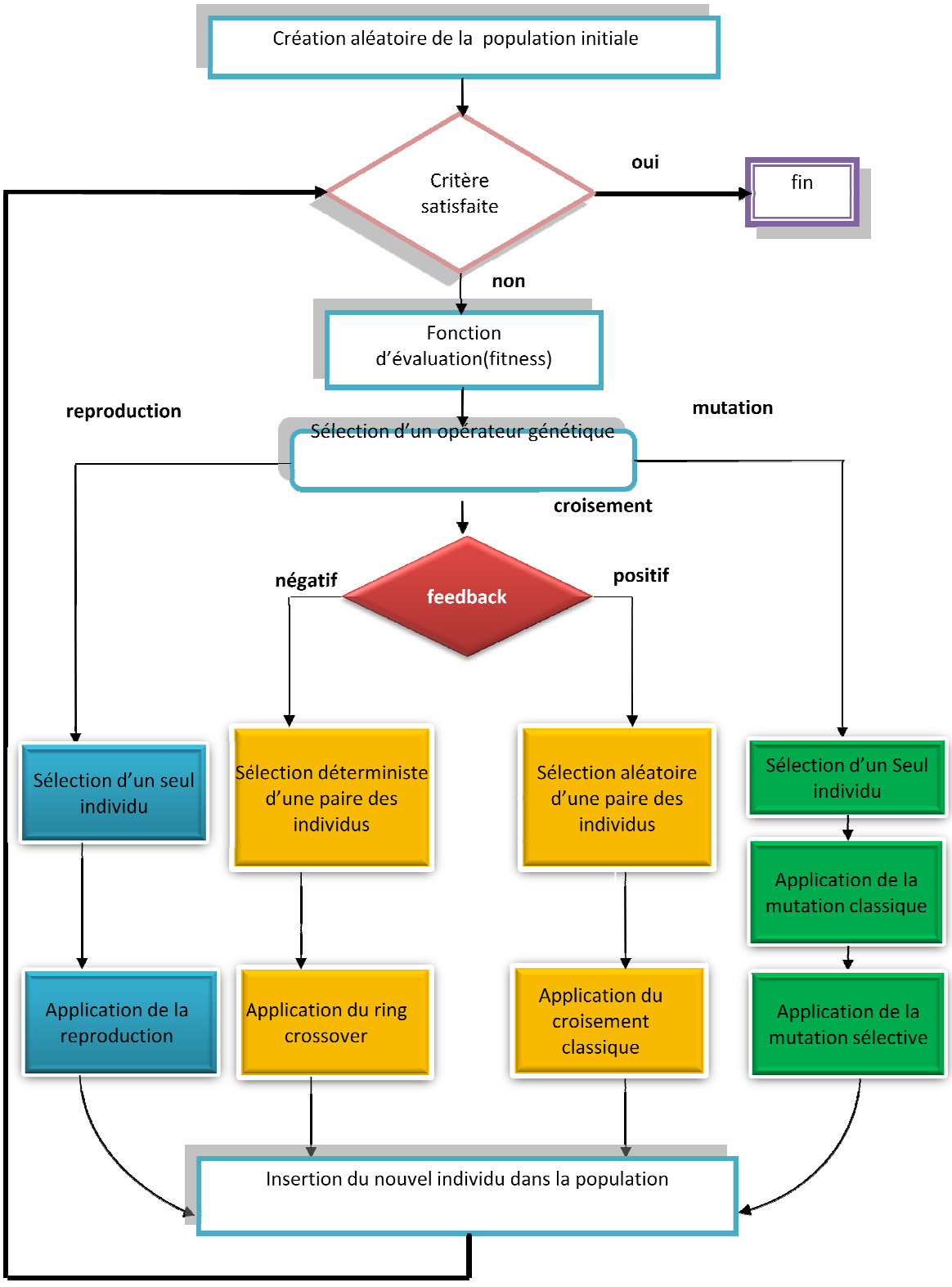

Figure 2.7 : Organigramme d'algorithme génétique adapté

2.6 Fonctionnement global

Suite à l'état perçu d'environnement, l'agent essaye d'élaborer la solution adéquate en interrogeant son module de performance, ce dernier fournit des connaissances au module d'apprentissage. Le processus d'apprentissage s'exécute en utilisant les propositions données par le générateur de problèmes, puis, une solution est générée, le module de performance s'occupe à l'exécuter. Selon la décision effectuée et son effet dans l'environnement, un rendement est renvoyé à l'agent, afin que le composant de critique estime un renforcement d'état, transmit ensuite au moteur d'apprentissage. Deux cas sont rencontrés, si le renforcement est positif, cela indique que l'apprentissage est efficace, et comme résultat, ce processus demeure selon les mêmes étapes ; utilisation du croisement classique c'est à dire, choisir de la population deux parents aléatoires, fixer un point de croisement et faire croiser les deux chaînes. Tandis que, si le renforcement est négatif alors le processus d'apprentissage sera modifié, en fait appel au nouvel opérateur de croisement cité auparavant.

2.7 Modélisation UML

La représentation d'ensemble des unités intervenant dans notre modèle est réalisée à l'aide l'outil UML intégré dans l'environnement EclipseUml2 pour modéliser les classes et leurs relations.

2.7.1 Diagrammes des agents

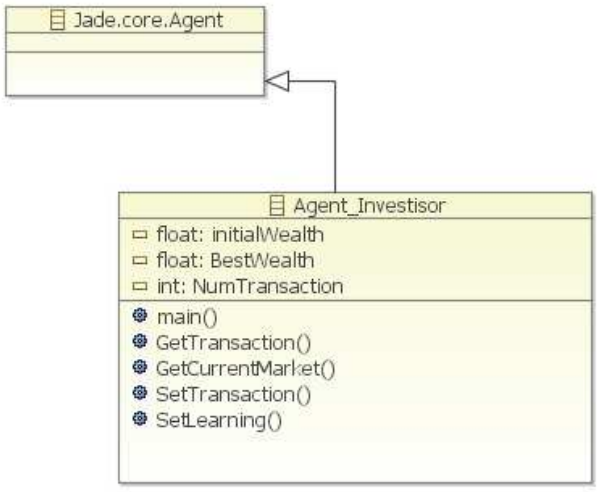

Figure 2.8 : Diagramme UML d'agent investisseur

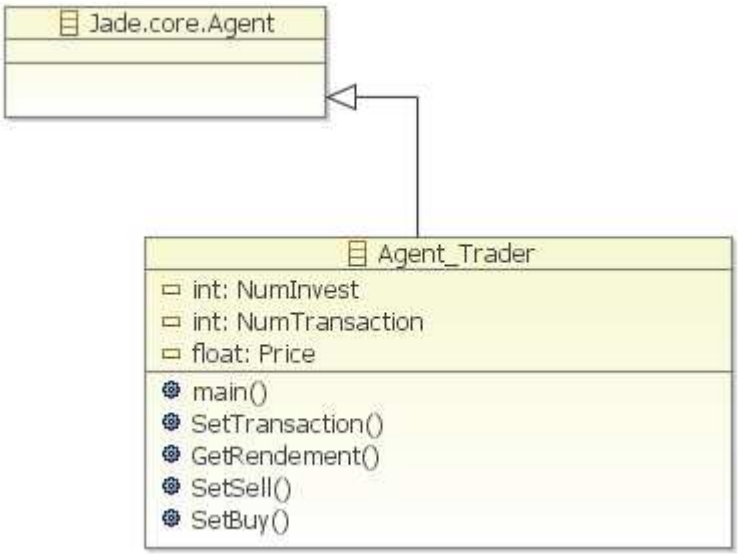

Figure 2.9 : Diagramme UML d'agent Trader

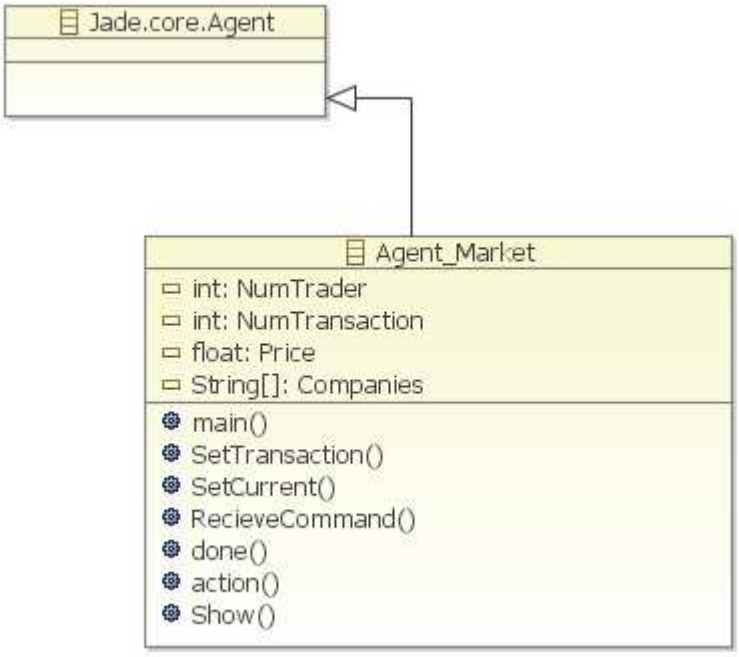

Figure 2.10 : Diagramme UML d'agent Market

2.7.2 Diagramme d'AG

En effet, l'algorithme génétique a comme but de faire évoluer une population de solutions dans le temps. Cette population est par le fait même constituée d'individus représentant ou comportant une partie de la solution possible au problème à résoudre. Un mécanisme de codage de ces individus est alors représenté par une classe Gene. Plusieurs

classes dérivent de ces trois classes abstraites, elles sont toutes illustrées dans le diagramme de classe de la figure 2.11.

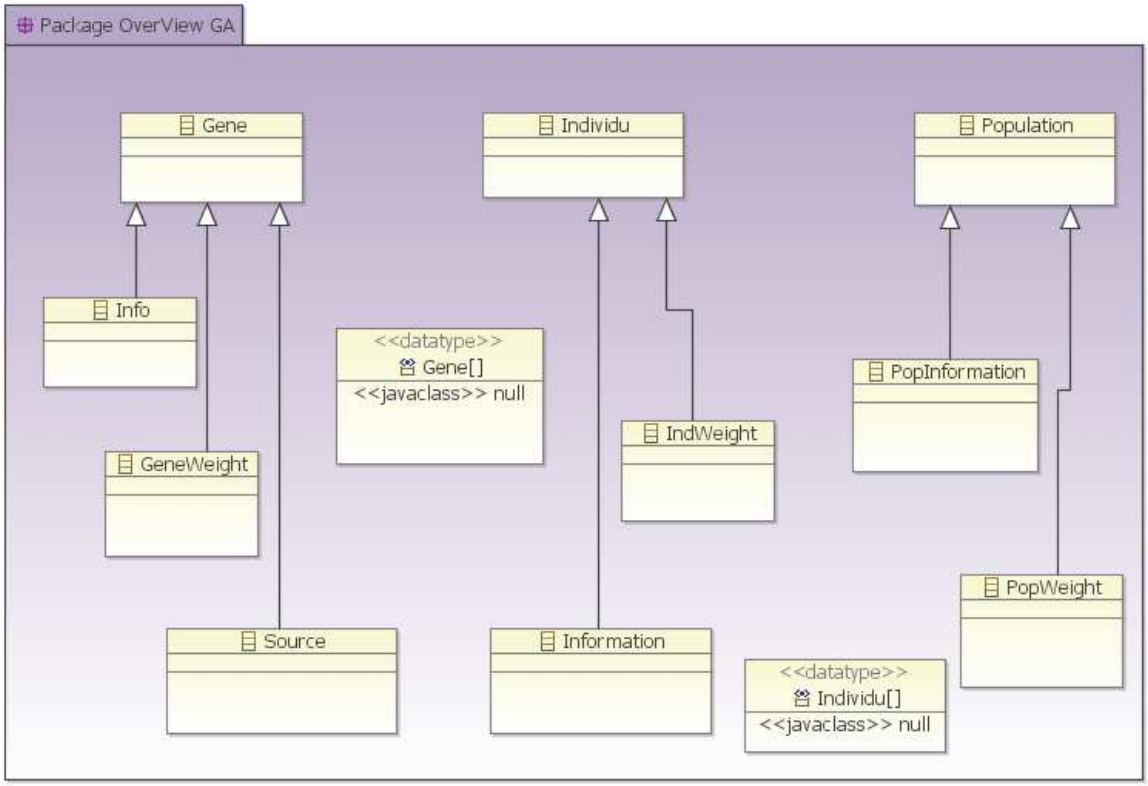

Figure 2.11 : Diagramme UML du package AG

L'implémentation d'AG a été développée à partir de trois classes qui constituent le fondement logique d'algorithme, Gene.java, Individu.java et Population.java

2.7.2.1 La classe Gene

La classe Gene (figure 2.12) représente l'élément de codage des individus qui composent la population. Cette classe ne comporte qu'une méthode abstraite de la mutation d'un gène. Généralement, la mutation est associée au gène et elle ne représente que la création aléatoire d'un gène. L'implémentation de la méthode de mutation se retrouve au niveau de l'individu

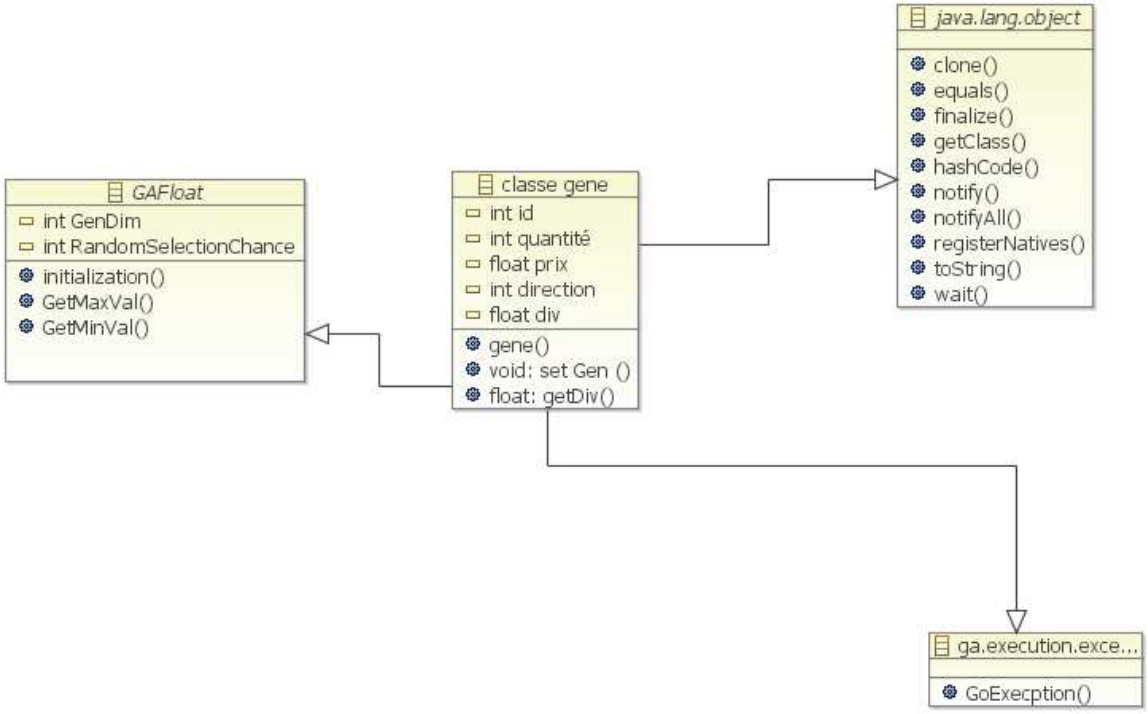

Figure 2.12 : Diagramme UML de la classe Gene

2.7.2.2 La classe Individu

La classe Individu (Figure 2.13) est représentée par un ensemble de gènes ainsi qu'une valeur d'adaptation qui caractérise l'individu. Les méthodes liées à la valeur d'adaptation de cette classe permettent de lire et d'écrire cette valeur de l'individu. D'autres méthodes permettent d'accéder à un des gènes spécifiques de l'individu, d'effectuer un croisement avec un autre individu, et de réaliser les mutations des gènes de l'individu. Les méthodes de croisement, de mutation et de calcul de la valeur d'adaptation sont des méthodes définies comme abstraites, car leur implémentation dépend du problème à résoudre. Il faut donc redéfinir les méthodes selon le cas. Par contre, on peut définir une certaine fonctionnalité générale.

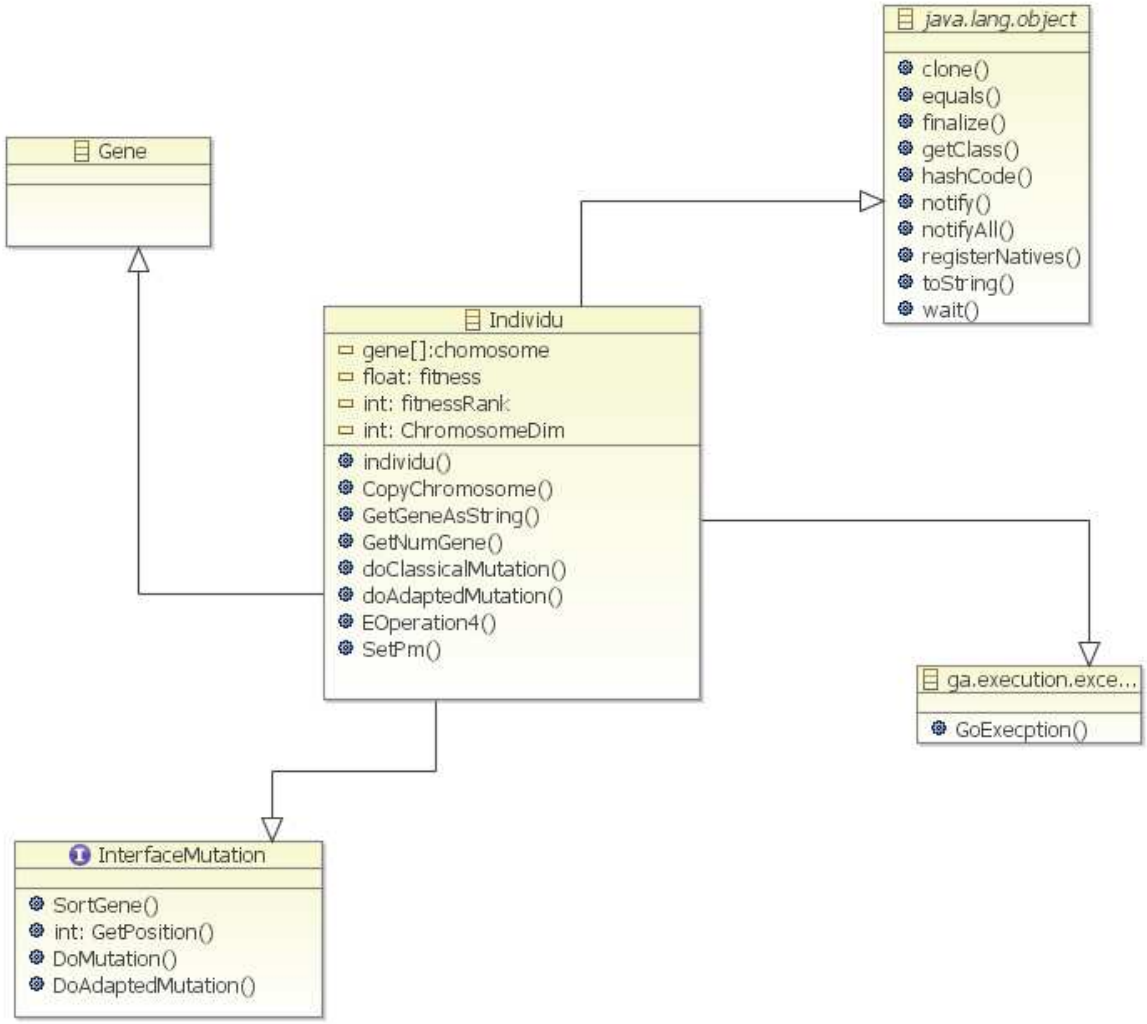

Figure 2.13 : Diagramme UML de la classe Individu

2.7.2.3 La classe Population

La classe Population (figure 2.14) permet de créer un ensemble d'individus et de les faire évoluer grâce aux opérateurs génétiques. La population que l'on crée possède plusieurs paramètres qui régissent son évolution. Ces paramètres sont détaillés dans l'applet et certaines méthodes permettent une modification de ces paramètres tandis que d'autres en définissent les constantes. La classe Population implémente donc toutes les méthodes qui permettent de définir ces paramètres de la population. Une fois que la population a été créée et que tous ces paramètres sont fixés, deux méthodes de cette classe permettent de passer d'une génération à la suivante. La plus simple effectue les étapes d'évolution suivantes :

- Croisement : On répète les étapes suivantes jusqu'à ce que le nombre d'individus soit augmenté dans les proportions fixées par le taux de reproduction. Puis, on tue autant d'individus, parmi les plus mauvais, qu'il est nécessaire pour revenir au nombre d'individus initial de la population.
- Mutation : On fait appel à la méthode mutation de chacun des individus en passant en paramètre la probabilité de mutation. Si l'individu est modifié par la mutation, on le range en fonction de sa nouvelle valeur d'adaptation.

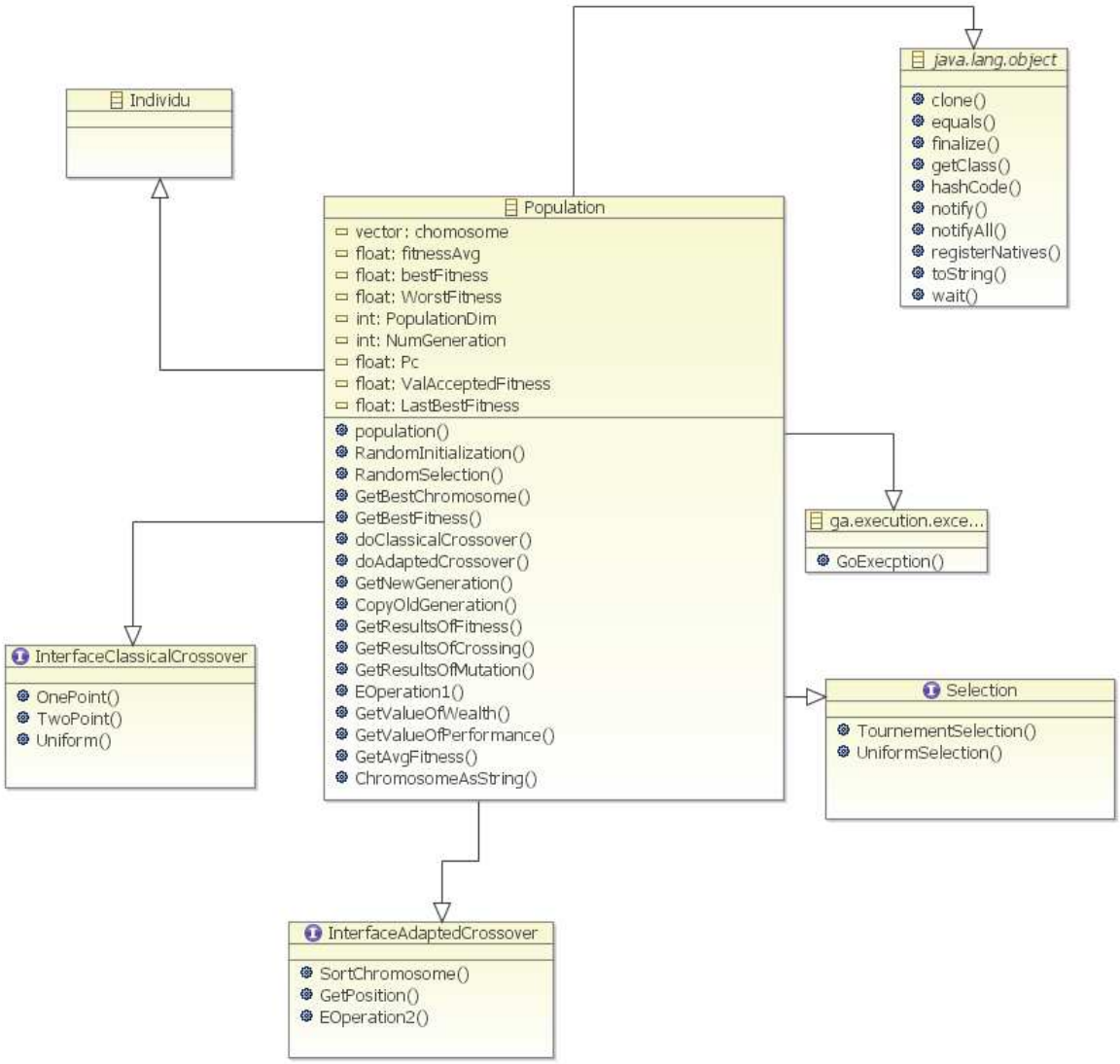

Figure 2.14 : Diagramme UML de la classe Population

Tous ces classes sont utilisées pour implémenté l'AG dédié à l'apprentissage, le diagramme UML correspondent est représenté dans la figure 2.15

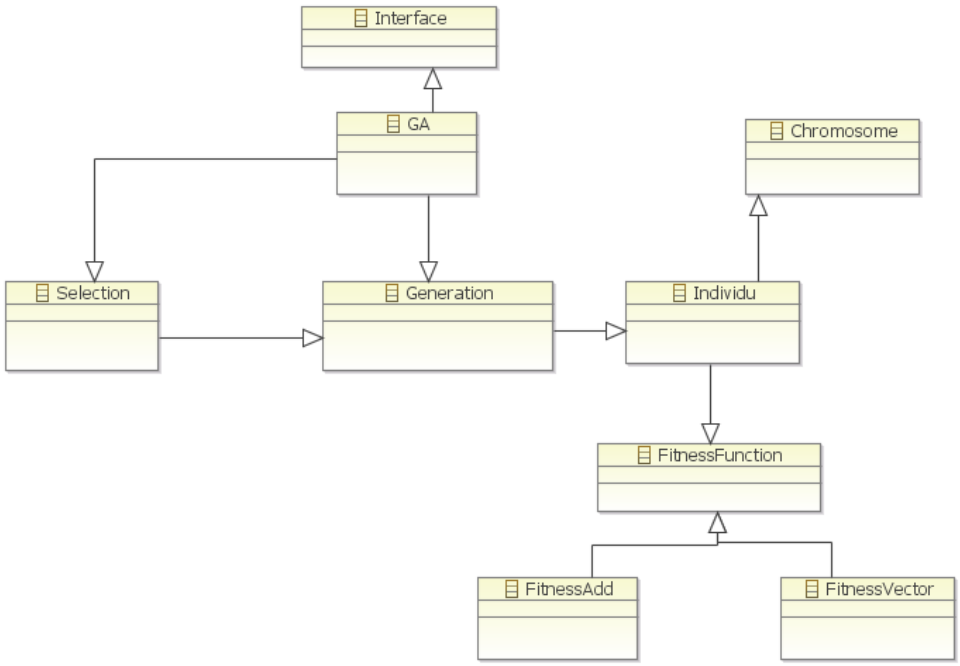

Figure 2.15 : Diagramme UML d'AG

2.7.3 Diagramme UML d'AR

Le protocole d'AR est implémenté à travers un ensemble des classes Java, ces classes peuvent être envisagées dans les figures 2.16, 2.17, 2.18 et 2.19 qui représentent les diagrammes UML correspondants.

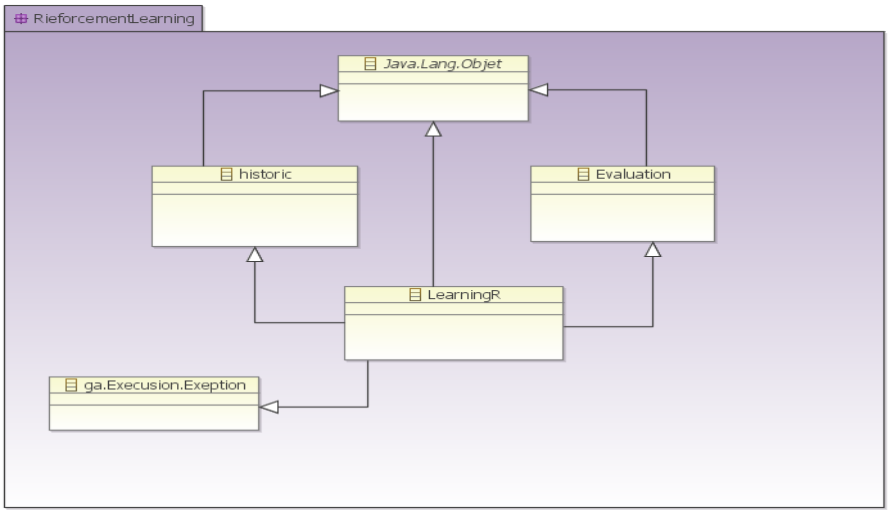

Figure 2.16 : Diagramme UML du package d'AR

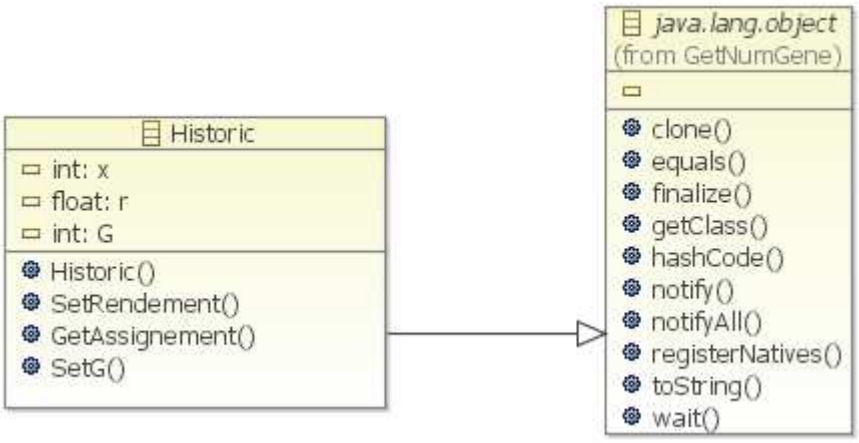

Figure 2.17 : Diagramme UML de la classe Historique

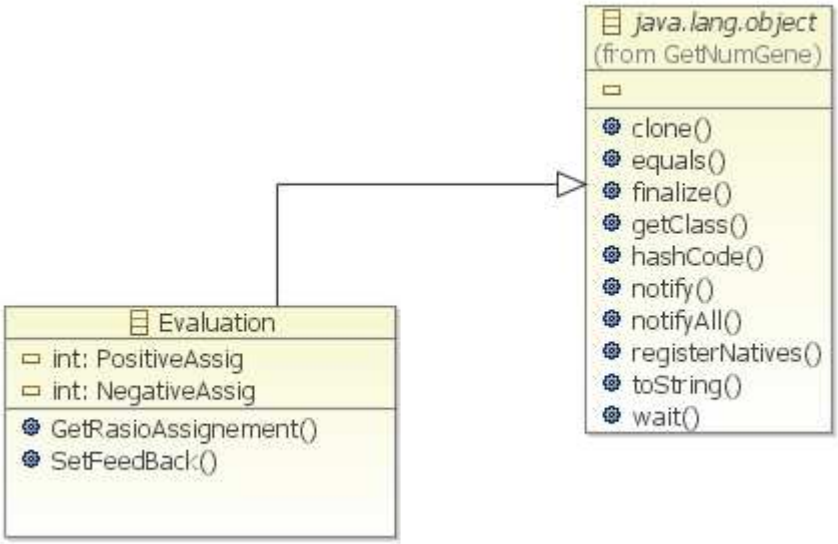

Figure 2.18 : Diagramme UML la classe Évaluation

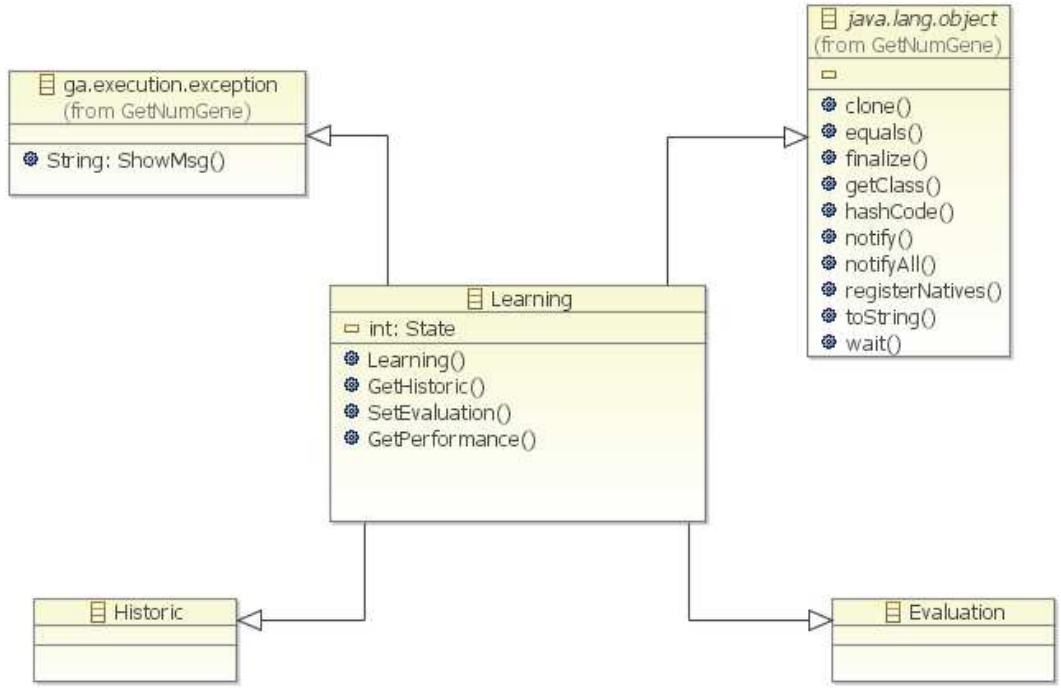

Figure 2.19 : Diagramme UML la classe Learning

2.8 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle architecture hybride d'un agent apprenant, fait intégrer un algorithme génétique, comme un outil d'apprentissage, permettant l'amélioration des performances d'un agent, et un apprentissage par renforcement qui assure le bon fonctionnement d'apprentissage, à travers l'assignement positif ou négatif des décisions effectuées dans l'environnement, auprès des solution choisis. Ces renforcements permettent de diriger l'élément d'apprentissage vers la bonne décision.

Afin d'implémenter ces deux processus, nous avons choisi pour chacun la stratégie qui convient. Pour évaluer les situations appris, nous avons utilisé une fonction d'évaluation basée sur le taux de récompense et pénalité. Cependant, un algorithme génétique adapté est proposé. Nous avons introduit deux alternatives de croisement et mutation dont l'objectif est d'augmenter la puissance d'AG en accélérant la vitesse de convergence vis-à-vis de l'évitement du phénomène connu sous le nom de convergence prématuré. Finalement une modélisation UML des différentes classes du système est présentée.

La partie qui suit est chargé à décrire un domaine d'application, notre choix s'est porté sur la bourse.

Chapitre 3

Étude de cas

Nous venons au long des précédents chapitres de présenter les algorithmes permettant de concevoir une architecture adopté pour qu'un agent peut apprendre. Dans le présent chapitre, nous présentons les résultats obtenus à travers un prototype ainsi qu'une étude de cas.

En effet, dans une première partie, nous décrivons le domaine d'application choisi : les marchés financiers, en présentant ses lois et ses principes.

Dans une seconde partie, nous proposons un scénarii illustrant les différents aspects de notre système, et évaluons les performances de ce dernier en terme de résultats produits et de temps d'exécution.

De plus, suite à l'analyse de ces résultats nous esquissons de propositions pour l'amélioration de performances de notre système.

3.1 Cadre du travail

Avant d'entamer notre étude, il est important de définir notre champ d'étude en termes d'environnement, de participant et de règle de jeu. Ces paramètres nous permettent de spécifier le contexte à travers lequel nous expliquons le choix et le principe de fonctionnement.

Dans notre travail nous somme intéressé aux marchés financiers comme domaine d'application qui sont détaillés dans la section suivante.

3.1.1 Structure des marchés financiers

3.1.1.1 Marchés financiers : ou la bourse de valeur

La bourse c'est le lieu où ils se rencontrent les demandes et les offres de capitaux, et l'échange des titres sous formes d'actions et obligations. Le fonctionnement du marché financier repose sur l'activité de deux compartiments dont les fonctions sont différentes et complémentaires : le marché primaire et le marché secondaire. Le marché primaire repose sur l'émission nouvelle d'un titre alors que le marché secondaire est celui où s'échange les titres déjà émis ***[83]***.

On distingue deux types de marchés, suivant comment interagissent les participants ***[83]***:

- Les marchés de contrepartie comme le NASDAQ(National Association of Dealers Automated Quotes System) qui sont des marchés continus décentralisés gouvernés par le prix où les teneurs de marché côtoient les investisseurs aux quels il s'échangent à vendre ou à acheter dans la limite de qualité spécifie à l'avance.
- Les marchés centralisés gouvernés par les ordres comme le NYSE(New York Stock Exchange), la bourse de Paris ou l'AMEX(l'Americain Stock Exchange) qui sont animés principalement par des sociétés de bourse qui transmettent les ordres des investisseurs finaux vers le marché.

Dans notre réalisation nous allons choisir de traiter l'échange des actions dans les marchés de contrepartie et principalement la bourse de NASDAQ.

3.1.1.2 Action

Une action est un titre de propriété délivré par une société de capitaux. Elle confère à son détenteur la propriété d'une partie de capital, avec des droits qui y sont associés :

intervenir dans la gestion de l'entreprise et en retirer un revenu appelé dividendes. Les firmes écoulant leur stock versent des dividendes (cash) D à chaque actionnaire qui dépend du succès de l'entreprise dans sa transaction. Les agents peuvent ainsi augmenter leurs profits à travers le courant des dividendes d'une part et à travers les spéculations émanant des fluctuations de prix de leurs actions d'autre part ***[83]***.

3.1.1.3 Les acteurs du marché

Ce sont les principaux acteurs des échanges sur le marché. Nous distinguerons d'une part les teneurs de marché qui assurent la liquidité permanente du marché et détenteurs d'actifs risqués dont ils fixent les cotations. Et d'autre part les investisseurs qui assurent la dynamique du marché d'achats en proposant des ordres d'achat ou des offres de ventes aux teneurs de marché qui fixent leurs prix selon le principe du plus offrant. Entre les teneurs de marché et les investisseurs s'appariaient les intermédiaires qui s'occupent à traiter les ordres des acteurs et terminer les transactions en toute sécurité (figure 3.1). En ayant fait le choix d'un marché de contrepartie gouverné par les prix, nous devons supposer les points suivants pour le fonctionnement du marché ***[83]***.

La dynamique des opérations suit le protocole suivant :

- Le tableau de prix affiche l'information publique, qui est le prix P de l'actif affiché sur le tableau pour tous les agents (investisseurs et teneur de marché).
- les investisseurs soumettent simultanément leurs ordres d'achat Ql ou de vente -Ql au teneur de marché.
- le teneur de marché révise son estimation de la valeur fondamentale M de l'actif considéré.
- le teneur de marché établit ses transactions et affiche ses cotations sous forme de meilleurs prix offerts et demandés à tous les agents demandeurs.
- le prix d'équilibre V: moyenne des meilleurs prix offerts et demandés est affiché sur le tableau de prix comme nouveau prix de tous les agents.

Nous suggérons que les ordres de ventes et d'achats d'un agent investisseur seront effectués selon le prix du marché.

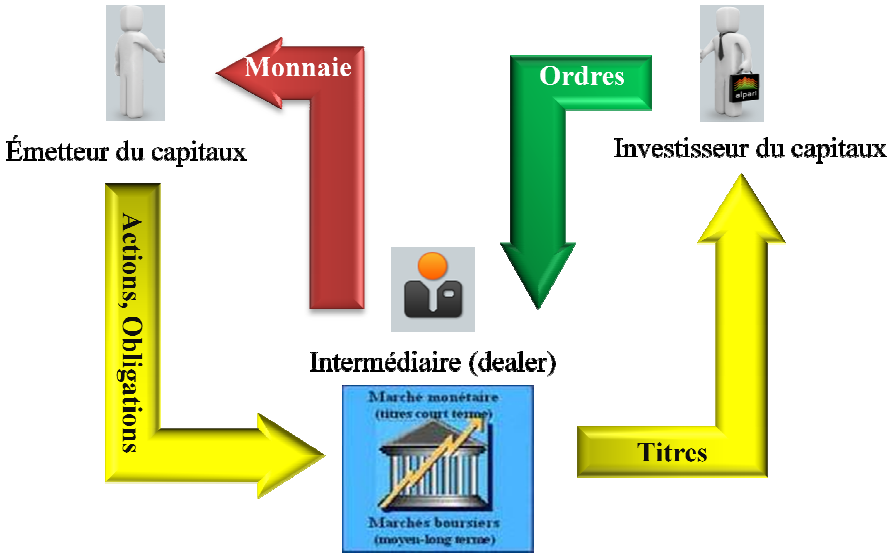

Figure 3.1 : Le modèle d'un marché financier

3.1.1.4 Les hypothèses d'un modèle de transmission par le prix

Les marchés financiers gouvernés par les prix se basent sur un ensemble des hypothèses dont elles détaillées comme suit :

H1 : le prix P est une variable aléatoire dont la distribution est normale, son espérance est nulle et sa variance s'écrit δ_p^2.

H2 : les agents sont constitués des investisseurs et des teneurs de marché ou trader. Un agent investisseur détermine ces actions, de manière à maximiser sa richesse finale, en tenant compte des informations fournis par son environnement, et des résultats des règles à succès des opérations précédentes.

H3 : les agents (investisseurs) ont une aversion au risque *ra* constante, ils ont donc une fonction d'utilité d'exponentielle négative.

$$U_i(W_i) = -e^{(-ra_i.w_i)} \qquad (3.1)$$

Équation 3 .1 : Calcul de la fonction d'utilité

Où W_i est la richesse finale du i-ème agent, U_i sa fonction d'utilité et ra_i son aversion au risque.

H4 : la richesse d'un agent investisseur est constitue d'un bien sous forme d'argent cash C_i et de réserve de stock d'actions h_i^α au temps t, des quantités Q_i^α d'actions peuvent

être vendues ou achetées contre de l'argent dans le marché financier au prix P affiché à l'instant t, pour une action h_i^α choisi. La richesse de chaque investisseur est donnée par :

$$W_i(t) = M_i(t) + Q_i^\alpha(t-1)[V(t) - P(t-1)] \quad (3.2)$$

Équation 3 .2 : Calcul de la richesse

Avec

$V(t)$: la valeur d'équilibre moyenne des meilleurs prix offerts et demandés est affiché par le tableau de prix, comme nouveau prix du marché.

$$M_i(t) = C_i(t) + \sum_{\alpha=1}^{k} h_i^\alpha.$$

H5 : la possession d'une action offre son propriétaire de bénéficier d'un montant d'argent (dividende). Ce rendement est une mesure de la croissance d'un capital, exprimé en pourcentage, sur une période de temps T = [t+1, t], s'écrit :

$$d = \frac{V_{t+1} - V_t}{V_t} \quad (3.3)$$

Équation 3 .3 : Calcul du rendement d'un titre financier

V_t : La valeur d'un titre en début de période.

V_{t+1} : La valeur d'un titre en fin de période.

H6 : un agent investisseur pendant la mise en œuvre de ses transactions dans le marché, risque de perdre une somme de sa richesse dû aux fluctuations de valeurs et l'incertitude des informations fournis.

H7 : l'estimation des rendements est calculée par leur moyenne historique, alors que le risque est estimé par le biais de l'écart-type des rendements.

H8 : les titres financiers peuvent être caractérisés par deux paramètres importants qui sont le risque et le rendement, ce couple est fortement corrélé. Un rendement élevé étant souvent synonyme à un risque important et vise versa. Un portefeuille diversifier peut se réduire le risque et augmenter le rendement des actions qu'il contient.

H9 : le rendement R_p d'un portefeuille correspond à la moyenne pondérée des rendements des titres individuels

$$R_p = \sum_{i=1}^{N} w_i d_i \qquad (3.4)$$

Équation 3 .4 : Calcul du rendement d'un portefeuille

w_i : la quantité de l'avoir i.

H10 : l'évaluation d'un portefeuille s'effectue par la comparaison du ratio de Sharpe, donné par la formule :

$$S_p = \frac{R_p}{\delta_p} \qquad (3.5)$$

Équation 3 .5 : Fonction d'évaluation d'un portefeuille

Avec :

S_p : Ratio de Sharpe du portefeuille étudié,

R_p : Rendement du portefeuille et

δ_p : écart-type du portefeuille (risque du portefeuille).

Il ressort immédiatement de cette formule qu'un ratio de Sharpe plus élevé est meilleur qu'un ratio de Sharpe bas.

3.1.2 Position du problème

Le problème essentiel de chaque agent investisseur est de choisir une combinaison d'actions à chaque instant t, moyennant un certain nombre de contraintes tel que le risque et le rendement. Son but est de maximiser sa richesse à chaque période de transition.
Nous pouvons formuler cet énoncé comme suit :

$$\begin{cases} \text{Maximiser} \quad R_p/\delta_p \\ \text{Avec :} \\ 0 \leq w_i < 1 \qquad \forall\, i \in \{1, \dots N\} \\ \sum_{i=1}^{N} w_i = 1 \end{cases} \tag{3.6}$$

Équation 3 .6 : Système d'équations de convergence

Si on note Q_i la quantité disponible d'une action *i*, sa proportion dans le portefeuille, w_i se calcule comme suit :

$$w_i = \frac{Q_i}{\sum_{i=1}^{N} Q_i} \tag{3.7}$$

Équation 3 .7 : Calcul du poids d'une action

L'objectif d'algorithme devient alors de maximiser une fonction non linéaire sous contraintes représentée selon l'équation 3.6.

3.2 Description d'AG

3.2.1 Représentation des individus

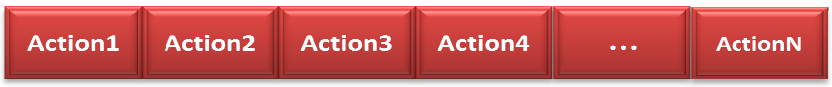

Figure 3.2 : Structure du chromosome

Le chromosome est constitué de N gène, chaque gène représente une action, et chaque action est caractérisée par cinq éléments :

- identificateur d'action : indique à quel secteur, entreprise appartient cette action.
- quantité d'action : indique le nombre d'actions que possède l'investisseur dans son portefeuille.

- l'opération à effectuer : soit acheter, vendre ou indifférent pou une action donné à un moment t.
- le prix : c'est le cours du titre.
- Le dividende : une valeur correspondante au bénéfice d'une action.

3.2.2 Codage

En générale, deux types de codage sont utilisés, un codage binaire et un codage réel. Nous utilisons une structure de données pour implémenter chaque individu.

3.2.3 Sélection

L'AG initialise aléatoirement pour chaque action une sous population puis la sélection par tournoi[⋆] permet de former l'ensemble des individus de la population initiale à partir des sous populations correspondant à différentes actions.

L'un des paramètres critique d'AG est la génération de la population initiale. La sélection par tournoi assure la diversité des éléments intervenant dans un individu, et le choix des w_i qui dépendent de tous les gènes (pas seulement de N_i) réduire la probabilité de tomber dans une solution non optimale.

3.2.4 Fonction du fitness

La fonction de mesure d'adaptabilité de la solution générée, consiste à calculer le ratio de Sharpe pour chaque chromosome (présenté précédemment), dont le but est d'optimiser ce ratio vers sa valeur maximale selon le système d'équation définie en 3.6.

3.2.5 Opérateur de mutation

L'opérateur de mutation est appliqué selon deux formes ; sa forme traditionnelle, puis la forme adaptée. Comme il est décrit précédemment, l'opérateur de mutation sélective part du point que la représentation des valeurs réel donne une chaîne de 0/1 triés de gauche à droite, du bit de poids fort vers le bit de poids le plus faible. Afin que notre chromosome suit le même principe et l'opérateur de mutation sélective sera applicable, on va trier les actions contenus dans chaque chromosome de gauche à droite selon leur dividende ; celui possédant le dividende le plus élevé sera placé au plus gauche et ce ayant le dividende le plus faible sera placé au plus droite du chromosome.

[⋆] Un tournoi consiste en une rencontre entre plusieurs individus pris au hasard dans la population. Le vainqueur du tournoi est l'individu de meilleure qualité ***[109]***.

3.2.6 Paramètres d'AG

3.2.6.1 Paramètres de la population

Les paramètres de la population doivent être fixés avant l'exécution de l'algorithme et ils ont une importance marquée dans la résolution du problème.

- **Nombre d'individus de la population**

Ce paramètre vise à fixer le nombre d'individus dans la population et ce pour toute la durée de l'exécution de l'algorithme. La taille de la population ne doit pas être trop grande car après une certaine limite la performance de l'algorithme diminue. En effet un nombre d'individus trop élevé affecte la rapidité de la résolution du problème.

- **Probabilité de croisement**

La probabilité de croisement (Pc) indique le taux de participation à la reproduction, soit la proportion de la population qui se reproduit par croisement. Si la probabilité de croisement est de 100%, alors toute la population participe au croisement. Par contre, si elle est de 0%, la nouvelle génération au complet est la copie exacte des individus de l'ancienne population.

- **Probabilité de mutation**

La probabilité de mutation P(m) indique la probabilité que chaque gène de chaque individu subisse une mutation lors d'une phase de reproduction. Si le taux de mutation est de 0%, les individus qui sont produits juste après le croisement ne comportent aucun changement. Par contre, si la probabilité de mutation est de 100%, tout le chromosome de l'individu est changé.

3.2.6.2 Paramètre des conditions d'arrêt

L'arrêt de l'algorithme dans la recherche de la meilleure solution intervient dès qu'un des paramètres atteint la valeur fixée.

- **Nombre de stagnations avant arrêt**

La stagnation se fait au niveau de la valeur d'adaptation du meilleur individu. Lorsque la valeur d'adaptation (fitness) du meilleur individu de la population se répète

constamment d'une génération à l'autre, on peut considérer que cette valeur est optimale et termine l'exécution de l'algorithme.

- **Valeur d'adaptation acceptable**

Ce test d'arrêt intervient lorsque l'on est capable de dire à partir de quelle limite on sera satisfait par la solution. Cela se traduira par la définition d'une valeur d'adaptation minimum de la solution (c'est le paramètre à donner). L'algorithme interrompra donc les recherches dès que la valeur d'adaptation du meilleur individu atteindra ou dépassera cette valeur.

- **Nombre maximum de générations**

C'est un nombre qui limite le nombre d'évolutions de la population générer par l'algorithme. La recherche est ainsi arrêtée après un certain nombre de générations.

3.3 Implémentation d'AR

Comme il a été détaillé auparavant, pour implémenter l'algorithme d'AR, on a besoin d'associer à chaque décision effectuée un renforcement. Dans le cas où notre agent est situé dans un marché financier, la méthode d'évaluer sa performance au fils du temps est la variation de sa richesse, à travers les transactions qu'il réalise.

Le bénéfice d'une opération est calculé par la formule :

$$b(t) = W(t) - W(t-1) \qquad (3.8)$$

Équation 3.8 : Calcul du bénéfice d'une opération

$W(t), W(t-1)$ représentent la richesse de l'agent investisseur au temps (t) et (t-1) successivement.

Le renforcement associé selon le bénéfice est obtenu comme suit :

$$\begin{cases} G(t) = 1 & si\ \ b(t) > 0 \\ G(t) = -1 & si\ \ b(t) < 0 \\ G(t) = 0 & si\ \ b(t) = 0 \end{cases}$$

3.4 Conclusion

Nous avons identifié dans ce chapitre le cas d'étude choisi, à travers la présentation des règles de base et les hypothèses correspondants. Ainsi que nous avons projeté notre proposition conceptuelle pour qu'il convient à cette application.

La section suivante sert à la réalisation du modèle proposé et l'interprétation des résultats obtenus.

Chapitre 4

Expérimentations et étude de performance

Après la spécification d'une architecture d'apprentissage par algorithme génétique, nous avons proposé un champ d'étude et l'ensemble des règles qui le gouvernent.

Ce chapitre est pour objectif à implémenter le modèle proposé, tout en analysant les résultats aboutis. Nous illustrant notamment l'effet des paramètres choisis sur la qualité des solutions trouvées.

4.1 Outils d'implémentation

Notre domaine d'application exige la réalisation d'un agent apprenant en interaction avec son environnement. Dans ce que suit nous détailleront les outils nécessaires pour accomplir cette tâche.

4.1.1 Environnement logiciel

JAVA est un langage orienté objet qui apporte un support efficace et élégant en définissant de manière très stricte la façon dont les objets communiquent entre eux. Le principal avantage est que chaque objet puisse être mis au point séparément ***[71, 75]***.

JAVA est extensible sans aucune limitation car pour étendre le langage il suffit de définir de nouvelles classes.

Contrairement à de nombreux compilateurs, celui de JAVA traduit le code source dans le langage d'une machine virtuelle (JVM). Ce code produit appelé « bytecode » est confié à un interpréteur qui le lit et l'exécute. C'est le rend portatif indépendamment du plate forme ***[71, 88]***.

Un autre avantage de ce langage de programmation réside dans le fait que la syntaxe de Java est analogue à celle de C++ ce qui le rend économique et professionnel, comme il est relativement simple à manipuler et ne nécessite pas un apprentissage ésotérique ***[88]***.

JAVA est un langage fiable, sécuritaire et polyvalent.

Plusieurs évaluations démontrent que Java, en termes de rapidité d'exécution, se rapproche et même quelques fois surpasse les performances d'applications équivalentes implémentées en C pur. De plus, le développement d'applications en Java est beaucoup plus rapide que la majorité des langages de programmation. Le choix était donc facile quant au langage utilisé pour l'environnement et la librairie de classes ***[88]***.

Un environnement de développement intégré (EDI ou IDE - Integrated Development Environment) est un programme regroupant un éditeur de texte, un compilateur, des outils automatiques de création, et souvent un débogueur. Pour le langage Java, il existe plusieurs EDI tels que NetBeans (de Sun), JBuilder (de Borland), JCreator ou Eclipse (d'IBM) ***[88]***.

Notre choix s'est porté sur Eclipse parce qu'il est écrit en Java, compatible à la plate-forme JADE choisie, ainsi qu'il permet la modélisation d'un système complexe composé d'un nombre infini d'entités autonomes situées dans un environnement statique ou dynamique, et fonctionnant en parallèle. En outre il est relativement simple à manipuler et possède des outils intégrés permettant un traitement aisé des résultats comme les illustrations graphiques***[88]***.

4.1.2 La plateforme multi-agents Jade

Les critères de choix d'une plateforme agent naturellement lié avec les besoins du système développé et ses caractéristiques, ainsi la plateforme devra répondre aux contraintes suivantes :

- La plateforme doit associer une méthodologie couvrant les différentes étapes du cycle de vie du développement d'un système multi-agents.
- La plateforme doit posséder une interface utilisateur pour faciliter le développement, ainsi que la réalisation des agents et leur gestion.
- Possibilité du suivie et de débogage.
- Facilité d'implémentation et de déploiement.
- La portabilité d'outil sur différent environnement.
- Disponibilité de documentation.
- Une plateforme libre.

Jade étant la plateforme proche de ces critères et nous l'avons utilisé pour implémenter l'architecture de notre système. C'est plateforme d'agent, développé par Gruppo telecom Italies.

Elle a pour but de simplifier le développement des agents tout en fournissant un ensemble complet des services et d'agent conforme aux spécifications FIPA dans un environnement JAVA. C'est pour cette raison que notre choix s'est porté sur cette plateforme.

Trois rôles principaux définissant une plateforme multi-agents sous la norme FIPA (figure 4.1) qui sont :

- le système de gestion d'agents (AMS - Agent Management System) : il fournit le service de nommage, assure que chaque agent possède un nom unique et représente la fabrication des

agents. En effet il peut créer et tuer un agent dans un conteneur de la plate-forme.

- le facilitateur d'annuaire (DF - Director Facilitator) : il fournit le service dit de «page jaune». Il a pour but d'aider à la recherche d'un agent grâce à son nom ou encore à ses compétences par exemple.
- le canal de communication entre agents (ACC – Agent Communication Channel) : il fournit la route pour les interactions de base entre les agents dans et hors de la plate-forme ; c'est la méthode de communication implicite qui offre un service fiable et précis pour le routage des messages.

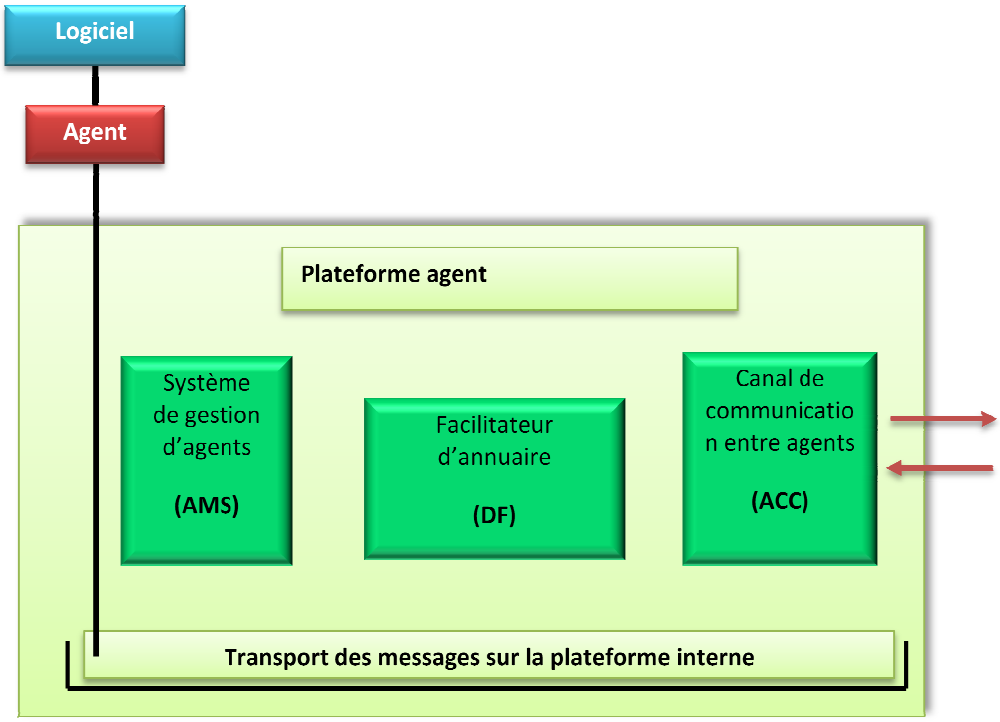

Figure 4.1 : Architecture d'une plateforme agent

La plateforme Jade fourni une interface graphique utilisateur(GUI) pour la gestion à distance des agents RMA (Remote Management Agent). Le GUI permet de créer, lancer un agent.

Les agents sont implémentés sous forme d'un Thread(agent schudeler) qui exécute à la suite et en boucle les comportements liés à l'agent(jusqu'à la fin).

La création d'un agent sous JADE passe par les 4 étapes suivantes, figure (4.2):

- faire hériter la classe de notre agent de la classe jade.core.Agent.
- définir les méthodes setup () et takeDown ().
- effectuer l'initialisation de l'agent dans la méthode setup () :
  - enregistrer l'agent dans le DF à partir de l'initialisation.
  - initialiser et lier les behaviours ("comportements") principaux de l'agent dans la méthode setup ().
- créer les behaviours spécifiques de l'agent.

Le package Jade.core implante le noyau du système. Il possède la classe agent qui doit être étendue par les applications des programmeurs. Une classe behaviour (comportement) est contenue dans jade.core.behaviours, une sous classe de jade.core. Les comportements implémentent les tâches ou intentions des agents.

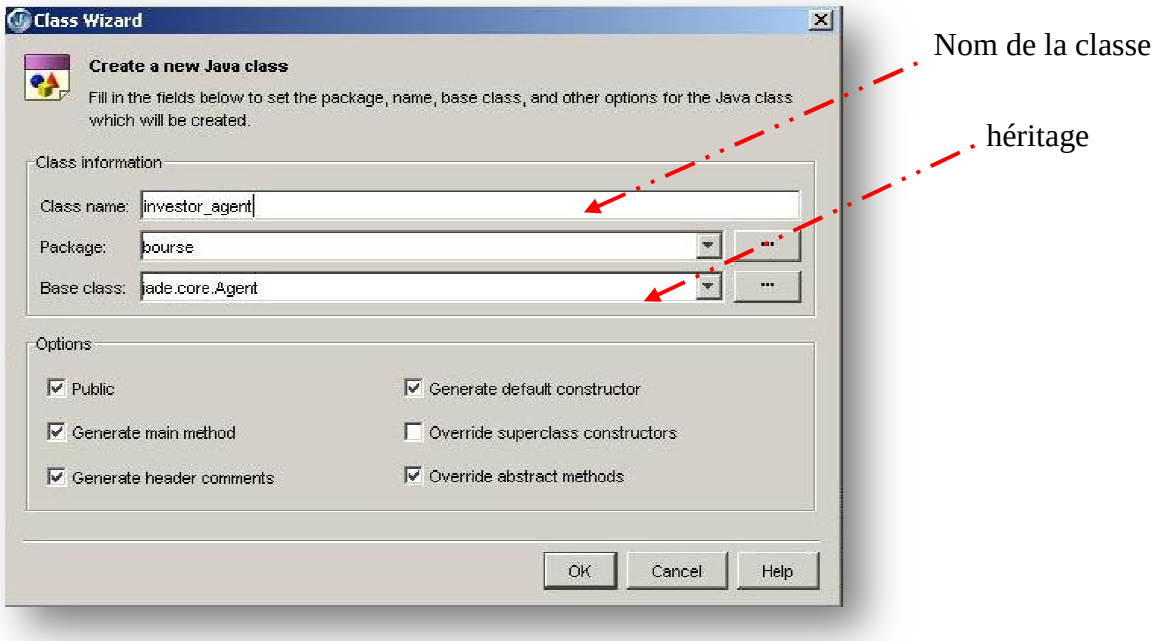

Figure 4.2 : Création d'une classe à partir de la super classe Agent.

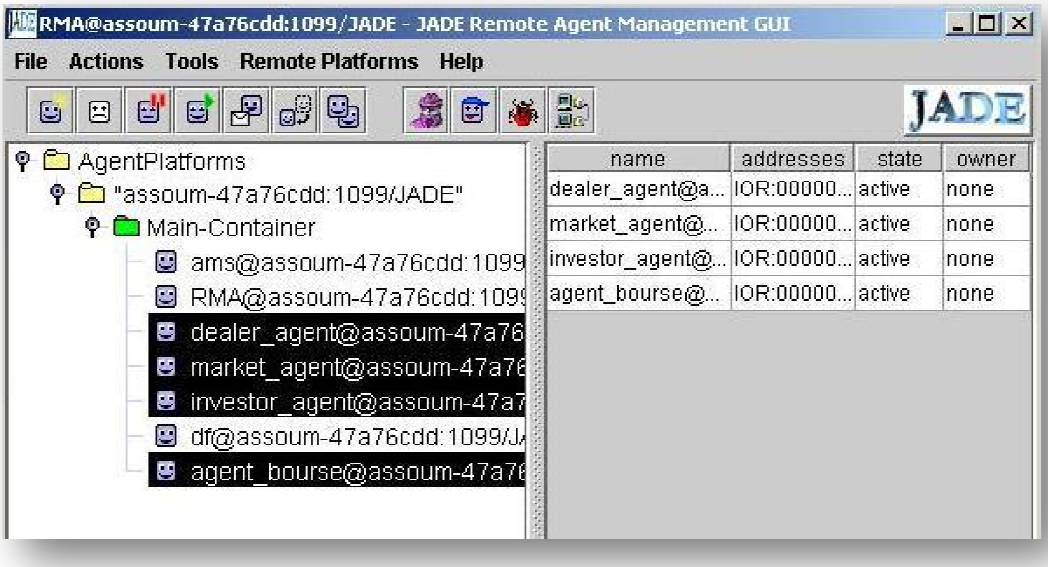

Figure 4.3 : Interface graphique de Jade contenant les agents créés.

Une fois l'agent est activé, il lui faut pouvoir communiquer avec les autres agents. Pour cela, on utilise le langage ACL(Acts Communication Language) de la norme FIPA. Le format de message est définit dans la classe jade.lang.acl.ACLMessage.

Lorsqu'un agent souhaite envoyer un message, il doit créer un nouvel objet ACLMessage, compléter ces champs avec des valeurs appropriées et enfin appeler la méthode send(). Lors qu'un agent souhaite recevoir un message, il doit employer la méthode receive() ou la méthode blockingReceive().

Voici certains des champs dont il dispose un message ACL :

Champs	*Description*
Sender	Expéditeur du message
Receiver	Destinataires du message
Content	Contenu du message respectant
Ontology	Ontologie utilisée par le contenu
Protocol	Protocole utilisé

Tableau 4 .1 : Les champs d'un message ACL

4.2 Les interfaces du système

Notre agent supposé d'être intéressé par l'investissement dans un marché financier (figure 4.4). Au début, son expérience initiative est minimale. Son but est d'augmenter sa performance au fil du temps, en profitant de ses pratiques passées pour qu'il acquière de l'expertise et apprenne les règles de succès.

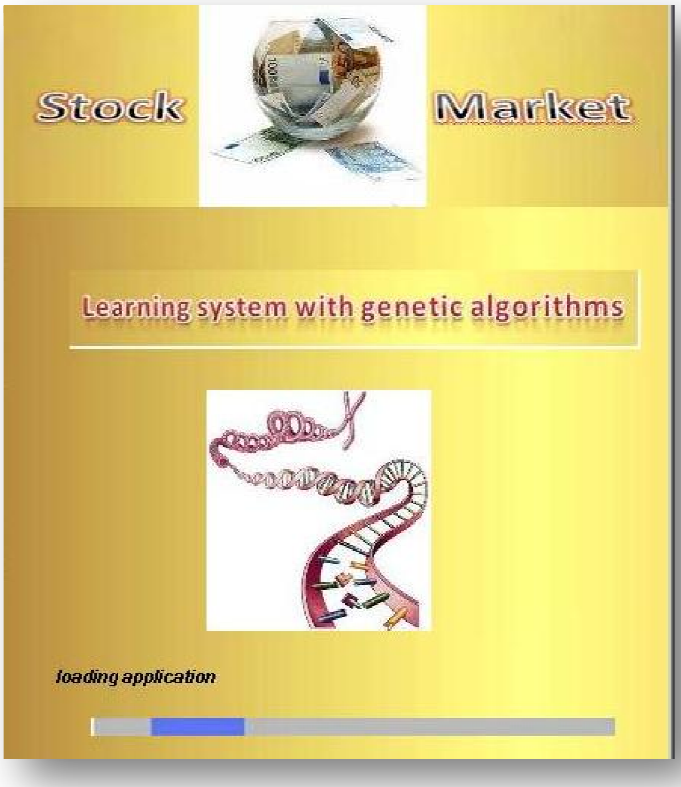

Figure4.4 : Interface de départ lors de l'exécution de l'application

La première étape à faire est l'inscription dans un site internet de la bourse NASDAQ, qu'à travers, l'investisseur peut suivre toutes les opérations et les actualités du marché, figure (4.5).

Figure4.5 : Interface d'authentification du système

Suite à l'inscription, l'investisseur est appelé à déterminer quelques paramètres nécessaires au démarrage, relatives au choix propre ; tel que le montant initial qu'il dépose et quel secteur économique veut il joindre, figure (4.6).

Figure 4.6 : Interface de configuration d'investisseur

Après l'ajustement du choix, l'investisseur peut commencer les sessions de ventes et d'achat. La figure (4.7) lui offre la possibilité de consulter les actualités du marché ainsi que l'état courant de son compte et l'historique de ses activités. À partir de ces informations, il cherche à trouver une transaction fructueuse selon son principe d'apprentissage.

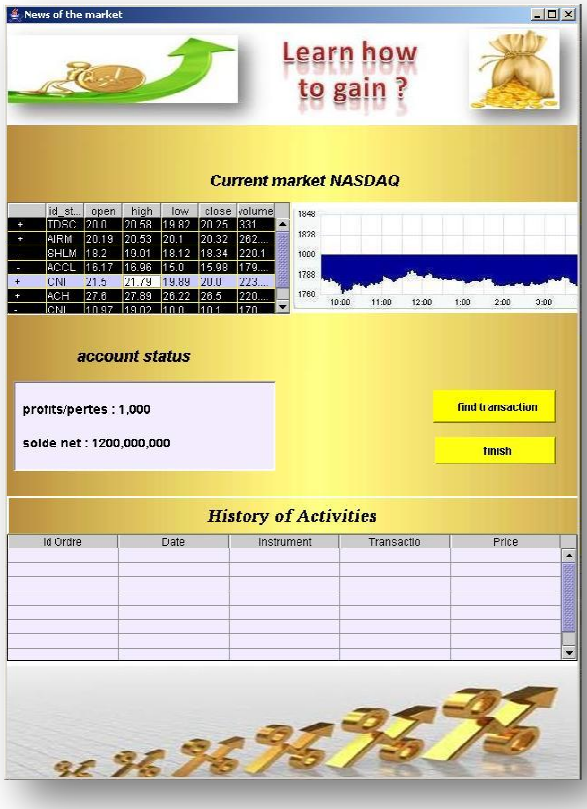

Figure 4.7 : Actualité du marché

À l'essor des données du marché, à un instant t, l'investisseur découvert plusieurs cas (solutions) parait adéquats, il lui faut choisir un ordre à effectuer. Une fois l'ordre est fixé, il est envoyé vers l'intermédiaire (dealer) concerné, figure (4.8).

Le cycle se revient autant que l'investisseur décide d'engager dans une nouvelle partie.

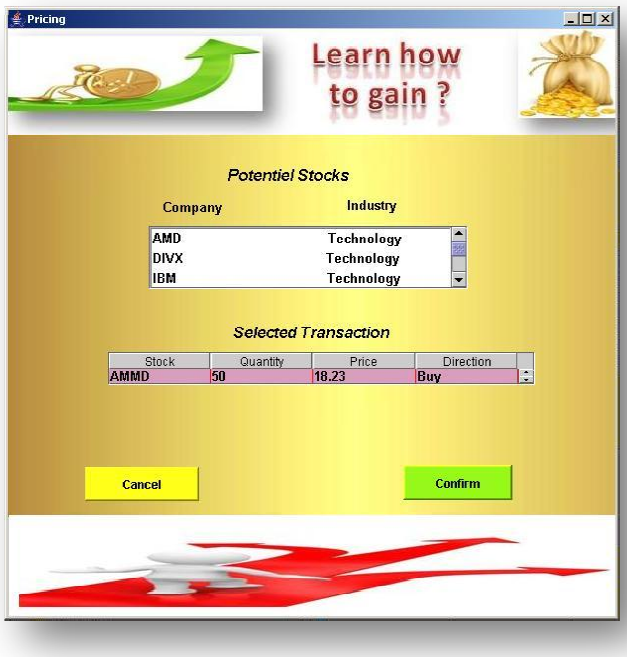

Figure 4.8: Recherche d'une transaction

Autant qu'il accomplisse des opérations, l'investisseur peut apprendre à partir de ses succès et échecs précédents à améliorer ses décisions et conclure des règles à appliquer dorénavant. « what have I learned » dans la figure 4.9 permet de visualiser le fruit de son apprentissage et donner le résumé d'exécution du programme en terme de fitness, générations et opérateurs d'AG (figure 4.10).

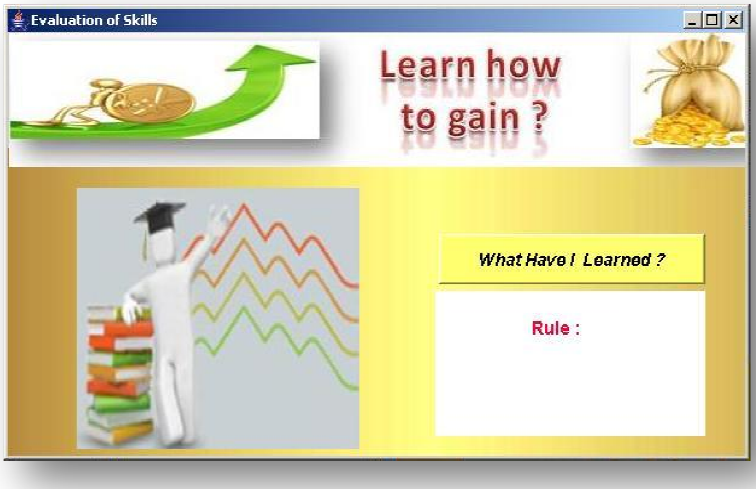

Figure 4.9 : Évaluation des performances

Les résultats d'apprentissage sont exprimés en termes des règles déduits automatiquement par l'agent, en utilisant les informations extraites de ses expériences.

Dans notre cas les valeurs enregistrées de la richesse obtenue suite à l'opération effectuée (vente ou achat), avec un prix P, correspondant à un ratio D/R (le dividende divisé par le taux d'intérêt de cette action), offrent à notre agent la possibilité de déduire que **« si le prix actuel de l'action est supérieur à D/R il sera préférable de vendre, et pour un prix inferieur à D/R, il sera préférable d'acheter »**.

Alors, « what have I learned » dans la figure 4.9, nous donne la règle suivante :

Si P > (D/R) alors vendre l'action i

Si P < (D/R) alors acheter l'action i

L'évolution des paramètres de l'AG est représentée par les courbes de son exécution, en appuyant le bouton « Draw Graph » figure 4.10.

Plusieurs types de sorties permettent de suivre en temps réel l'exécution des algorithmes, comme le montre la figure 4.10. Ces sorties permettent ainsi de comparer la performance des algorithmes pour la résolution d'un même problème.

- Fitness maximum : Affichage de la valeur d'adaptation du meilleur individu de la population courante.
- Richesse correspondante : Affichage de la richesse maximale, ce qui représente la solution trouvée par les algorithmes génétiques.
- Nombre de générations : Affichage du nombre de générations parcourues par l'algorithme.
- Temps de calcul : Affichage du temps de calcul pour trouver une solution.
- Graphique des courbes de la valeur d'adaptation : Affichage du dessin des courbes de la meilleure valeur d'adaptation et de la moyenne des valeurs d'adaptation de toutes les populations en fonction du nombre de générations.

4.2 EXPÉRIMENTATION

ERROR: ioerror
OFFENDING COMMAND: image

STACK:

Bibliographie

[1] J.Arifovic, M.K. Mascheky. « Revisiting Individual Evolutionary Learning in the cobweb Model - An Illustration of the Virtual Spite-E» , 2006.

[2] S.Aupetit, N.Monmarché, M.Slimane. “Hidden Markov Models Training Using Population-based Metaheuristics”, Natural Computing Series Springer, 2008.

[3] S.Aupetit, N.Monmarché, M.Slimane. « Utilisation des Modèles de Markov cachés pour la reconnaissance robuste d'images : apprentissage par colonie de fourmis, algorithme génétique et essaim particulaire ». Optimisation en traitement du signal et de l'image, Hermès-Lavoisier, 2007.

[4] M.Baccouche, J.Boukarouche, M.Benaissa, A.Benabdelhafid. « Étude Comparative des Méthodes de Sélection dans les Algorithmes Génétiques », Premier Congrès International de Signaux, Circuits & Systèmes, Tunisie, 2004.

[5] L.Baird, A.Moore. « Gradient descent for general reinforcement learning ». In NIPS, 1998.

[6] N.Barnier, P.Brisset. « Optimisation par hybridation d'un CSP avec un algorithme génétique », JFPLC, 1997.

[7] N.Ben Amor, S.Benferhat, Z.Elouedi. « Réseaux bayésiens naïfs et arbres de décision dans les systèmes de détection d’intrusions », RSTI-TSI. Volume 25, pp 167-196, 2006.

[8] Benachour. « Modélisation et apprentissage d'agents artificiels adaptés à un marché financier », thèse de doctorat, université de Paris 12 ,2000.

[9] D.Billings. « Thoughts on roshambo ». ICGA Journal, 23(1):3–8, 2000

[10] G.Bisson. « La similarité : une notion symbolique / numérique », 4èmes journées sur l' «induction symbolique/numérique», Orsay, pp 93-96, 1994.

[11] G.Blanchard. « Une brève introduction au support vecteur machines », Journées Ouvertes Biologie Informatique Mathématiques. Rennes, Berlin, 2005.

[12] L.Booker, D.E. Goldberg,J.Holland. « Classifier Systems and Genetic Algorithms». Artifcielle Intelligence. 40(1-3): 235-282.1989.

[13] R.Brafman, M.Tennenholtz. « R-max - a general polynomial time algorithm for near-optimal reinforcement learning». In IJCAI, 2001.

[14] Y .Braouezec. « Apprentissage et conflit exploration-exploitation : un essai », 2004.

[15] C.Buche. « Apprentissage par l'action pour les comportements d'agents autonomes »,2002.

[16] V.Camps, M.Gleizes. « Réflexions sur l'apprentissage en univers multi-agents », Journée du PRC GDR IA "Les systèmes multi-agents", Toulouse, Editions Hermès, 1997.

[17] L.Candillier, I. Tellier, F. Torre, O. Bousquet. « Évaluation en cascade d'algorithmes de clustering », 2èmes Rencontres Inter-Associations sur la classification et ses applications RIAs, 2006.

[18] A. Cardon, T.Galinho, J-P.Vacher. « Genetic Algorithms using Multi-Objectives in a Multi-Agent System», Robotics and Autonomous Systems, 33(2-3):179-190, 2000.

[19] B.Chen, M.Haddar, O.Kaf, Grégois M.Montecheuil, M.El Beze, « Contexte multilingues alignés pour la désambiguïsation sémantique : une étude expérimentale ». TALN Dourdam, 2005.

[20] T.Conde. « Méthodes infographique pour l'apprentissage des agents autonomes virtuels », thèse doctorat, université Lausanne, Suisse, 2005.

[21] V.Conitzer and T.Sandholm. « AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents». In ICML, pp 83–90, 2003.

[22] P.Cornic, A.Michel. « De la création d'images composites à la par coopération d'algorithmes en contexte multicapteurs », Traitement du Signal — Volume 12 - n° 5. 1995.

[23] A.Cornuéjols, L.Miclet. « Apprentissage artificiel et robotique : une introduction », Journées Nationales de la Recherche en Robotique, Murol, pp 121-126,2003.

[24] A.Cornuéjols, L.Miclet. « Les arbres de décision et leur apprentissage, Apprentissage Artificiel : Méthodes et Algorithmes », chapitre 11, Eyrolles, 2002.

[25] A.Cornuéjols, L.Miclet. « What is the place of machine learning between pattern recognition and optimization?», TML-2008 Conference Teaching Machine Learning, France,2008.

[26] A.Cornuéjols. « Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à vaste marge ». Bulletin de l'A F I A, 2002.

[27] K.De Jong. « The analysis of the behaviour of class of genetic adaptative systems», thèse de doctorat, university of Michigan, 1975.

[28] K.De Jong. « Learning with Genetic Algorithms: An Overview, Machine Learning»3(2):121-138, 1988.

[29] K.De Jong, W.Spears. « On the virtues of parameterised uniform crossover». In R. K. Belew and L. B. Booker, editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pp 230-236, San Mateo, July 1991.

[30] K.De Jong, J.Sarma. « On Decentralizing Selection Algorithms». Proceedings of the Sixth International Conference on GeneticAlgorithms. 17-23. Morgan Kaufmann. 1995.

[31] D.Defays, Robert M. French , J. Sougné. « Apports de l'Intelligence Artificielle à la Psychologie », In J, Rondal (Ed), Introduction à la psychologie scientifique, pp 379-415, 1999.

[32] F.Delorme, G.Loosli. « Un outil générique pour l'analyse automatique et la visualisation de productions d'apprenants », journal TICE, 2006.

[33] J.L.Deneubourg, S.Gross, R.Beckers, G.Sandini. « Collectively self-solving problems, in Self organization, emergent properties and learning». A. Babloyantz editors, Plenum, 1991.

[34] B.Drieu. « L'intelligence artificielle distribuée appliquée aux jeux d'équipe situés dans un milieu dynamique : l'exemple de la RoboCup », université Paris 8,2001.

[35] A.Dutech, R. Aras, F. Charpillet. « Coordination par les jeux stochas-tiques ». In CARI,2006.

[36] X.L.Fàbrega, J.M.Genifer. « A Nearest Neighbour based Classifier System using GA», Proceeding of the Genetic and Evolutionary Computation Conference (GECCO99), 1999.

[37] J.Ferber. « Les systèmes multi-agents:un aperçu général. Techniques et sciences informatiques ». Vol. 16, No. 8, pp 979-1012, 1997.

[38] J.Ferber. « Multi-Agent System: An Introduction to Distributed Artificial Intelligence, Harlow», Addison Wesley Longman, 1999.

[39] L.Gacôgne. « Apprentissage génétique global d'un contrôleur flou a deux variables basé sur la simulation d'un véhicule autonome », IPMU, pp 1099-1104, 1994.

[40] P.Gallinari, H.Zaragoza, M.Amini, « Apprentissage et données textuelles », CAp, Paris,2000.

[41] P.Gaussier. « Learning and communication via Imitation: An Autonomous Robot Perspective », 2001.

[42] P.Gaussier, S. Moga, M. Quoy. « From perception-action loops to imitation processes:A bottom-up approach of learning by imitation», Robotics and autonomous systems, AAI, 1998.

[43] S.Gelly, N.Bredeche, M.Sebag. « Inférence dans les HMM hiérarchiques et factorisés : changement de représentation vers le formalisme des Réseaux Bayésien », Actes de l'Atelier sur les Modèles Graphiques Probabilistes, France, 2005.

[44] E.D.Goldberg. « Genetic Algorithms in Search, Optimization, and Machine Learning». Addison-Wesley Publishing company 1989.

[45] E.D.Goldberg. « Real-coded Algorithms, virtuel Alphabets, and Blocking», University of Illinois at Urbana Champaign, 1990.

[46] N.Gomond, J.M.Salotti. « Extended model of conditioned learning within latent inhibition». ESANN, pp 88-94, 2006.

[47] B.Gosselin. « Classification et reconnaissance de forme », 2000.

[48] J.Grefenstette. « Genetic Algorithms and Machine Learning», Invited talk, Sixième ACM Conférence. Computational Learning Theory (COLT 93), 1993.

[49] J. Grefenstette. « Genetic Algorithms and Their Applications», Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ: Lawrence Erlbaum, 1987.

[50] Y.Guermeur. « SVM Multiclasses, théorie et applications », habilitation à d'ériger des recherches de l'université Henry Poincaré, France, 2007.

[51] S.Guillas, K.Bertet, J.M.Ogier. « Reconnaissance de symboles bruités à l'aide d'un treillis de Galois », CIFED, pp 85-90, Fribourg, Suisse, 2006.

[52] H.Hadj-Mabrouk. « Apport du raisonnement à partir de cas à l analyse de la sécurité des logiciels dans les transports ferroviaires », 3rd International Conférence: Sciences of Electronic, Technologies of Information and Telecommunications Tunisia, 2005.

[53] M.Haned, S.A.Addouch, A.El Mhamedi. « Construction d'un modèle de prédiction de l'impact d'un état de performance sur la performance globale d'un processus en utilisant les réseaux Bayésiens », CPI'2007 Rabat, Maroc, pp 1-14, 2007.

[54] J.k.Hao, P.Galinier, M.Habib. « Méta-heuristiques pour l'optimisation combinatoire et l'affectation sous contraites », Revue d'Itelligence Artificielle, 13(2) :283-324., 1999.

[55] J.Haton. « L'intelligence artificielle », 263eme conférence de l'Université de tous les savoirs donnée, 2000.

[56] M.Hibou. « Réseaux Bayésiens pour la modélisation de l'apprenant en EIAH : modèles multiples versus modèle unique », 1ères Rencontres Jeunes Chercheurs en EIAH, pp 40-47, 2006.

[57] J.Holland. « Outline for a logical theory of adaptative systems». Journal of the association of computing machinery,3,1962.

[58] J.Holland. « Adaptation in Natural and Articial Systems», Universityof Michigan Press, Ann Arbor. 1975.

[59] J.H.Holmes, P.L.Lanzi, W.Stolzmann, S.W.Wilson. « Learning Classifier Systems: New Models», Successful Applications, Inf. Process. Lett., 82(1): 23-30, 2002.

[60] J.Hu, M. Wellman. « Multi-agent reinforcement learning: theoretical framework and an algorithm». In ICML, pp 242–250, 1998.

[61] N.R.Jennings,M. Wooldridge. « Intelligent agents: Theory and practice», The Knowledge Engineering Review, 10(2), pp 115-152, 1995.

[62] S.H.Jung. « Selective Mutation for Genetic Algorithms», World Academy of cience,Engineering and Technology 56, 2009.

[63] A.Just, M.Sébastien, O.Bernier, J.Viallet. « Reconnaissance de gestes 3D bi-manuels Two Handed 3D Gesture Recognition », Workshop Acquisition du geste humain par vision artificielle, RFIA , 14ème Congrès Francophone, Toulouse, France, 2004.

[64] I.Kallel, A.M. Alimi. « MAGAD-BFS: A learning method for Beta fuzzy systems based on a multi-agent genetic algorithm », 2006.

[65] M.Kearns, S. Singh. « Near-optimal reinforcement learning in polynomial time». In ICML,1998

[66] R.Kessler, J.M.Torres-Moreno, M.El-Beze. « Classification thématique de courriels avec apprentissage supervisé, semi supervisé et non supervisé ». VSST, Toulouse, pp 493-504,2004.

[67] L.Knight, S.Sen. «PLEASE: A Prototype Learning System Using Genetic Algorithms», ICGA, pp 429-435, 1995.

[68] M.Kuri. «An Alternative Model of Genetic Algorithms as Learning Machines» ,Expert Systems with Applications, Volume 15, Issues 3-4, pp 351-356. 1998.

[69] E.Kutchinski, T. Uthmann, D. Polani. «Learning competitive pricing strategies by multi-agent reinforcement learning», Journal of Economic Dynamics and Control, 27: pp 2207-2218, 2003.

[70] T.Laloe. « Une approche de type K-plus proches voisins pour la regression fonctionnelle », 41èmes Journées de Statistique, SFdS, Bordeaux 2009.

[71] H.Larochelle. « Étude de techniques d'apprentissage non-supervisé pour l'amélioration de l'entraînement supervisé de modèles connexionnistes », thèse de doctorat PHD, Université de Montréal, 2008.

[72] P.Leray. « Réseaux bayésiens : apprentissage et modélisation de systèmes complexes », habilitation à diriger les recherches, INSA, Rouen, 2006.

[73] P.Lucidarme, A. Liégeois, J.L. Vercher, R. Bootsma. « Un algorithme évolutionniste pour l'auto-apprentissage de groupes de robots mobiles autonomes », NSI'02, La Londe des Maures, article 56, Session Modèles 2, 2002.

[74] T.MTri Do, T.Artières. « Optimisation du Primal pour les SVM », Extraction et Gestion des Connaissances (EGC), Nice, 2008.

[75] M.Métivier. « Méthodes évolutionnaires et apprentissage : Apprentissage par imitation dans le cadre des systèmes de Classeurs », thèse de doctorat, université Paris 5, 2004.

[76] R.S.Michalski, Kodratoff. « Machine Learning-An artificial Intellingence Approach» Vol-III, Michalski & Kodratoff Eds, Morgan Kaufmann, 1990.

[77] R.S.Michalski. « Learning strategies and automated knowledge aquisition: an over view»,1984.

[78] R.S.Michalski, R.Stepp. « Automated construction of classifications conceptual clustering versus numerical taxonomy», IEEE.trans, on pattern analysis and machine learning, volume PAMI-5, NO.4, 1983.

[79] S.Moga. « Apprendre par imitation : une nouvelle voie d'apprentissage pour les robots autonomes », thèse de doctorat, université de Cergy-Pontoise.e, 2000.

[80] R.Neapolitan. «Learning bayesian networks», Prentice Hall, 2003.

[81] A.Newell, A.Newell. «The knowledge level», Artificial Intelligence 18, pp 87-127, 1982.

[82] M.Nguyen. « Contrôle des interactions orales entre humain et machine : approche d'apprentissage machine », thèse de doctorat, université du Québec, Canada, 2007.

[83] P.Navatte, J.C.Augros. « Bourse les options négociables », vuibert gestion, collection dirigée par j-p.helfer et j.orsoni, 1987.

[84] F.Olivier. « De l'identification de structure de réseaux bayésiens à la reconnaissance de formes à partir d'informations complètes ou incomplètes », thèse de doctorat, l'Institut National des Sciences Appliquées de Rouen, 2006.

[85] F.Olivier, Philippe Leray. « Étude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens », JEDAI, 2006.

[86] F.Olivier, PLeray. « Apprentissage de structure des réseaux bayésiens et données incomplètes », Revue des Nouvelles Technologies de l'Information, RNTI-E-3, 2005.

[87] C.Parthenay, H.Simon. « Rationalité limitée, théorie des organisations et sciences de l'artificiel ». Ed Management & Société, 2008.

[88] P.Pellerin. « Méta-apprentissage des algorithmes génétiques », thèse de doctorat, Université du Québec, Canada, 2005.

[89] S.Perret. « Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure », thèse de Doctorat. Université Joseph Fourier - Grenoble I, 1997.

[90] T.Rahwan, R.Ashri. « Agent-Based Support for Mobile Users Using AgentSpeak». In: Giorgini P., Henderson-Sellers B., Winikoff, M. (eds.): Proceedings of the Workshop on Agent-Oriented Information Systems AOIS, Australia, 2003.

[91] R.Rakotomalala. « Arbres de décision », Revue Modulad, Numéro 33, 2005.

[92] L.Remaki, J.Guy. Meunier. « Un modèle HMM pour la détection des mots composés dans un corpus textuel ». JADT : 5es Journées Internationales d'Analyse Statistique des Données Textuelles, université du Québec, Canada, 2000.

[93] M.F.Rousseaux. « Une contribution de l'intelligence artificielle et de l'apprentissage symbolique automatique à l'élaboration d'un modèle d'enseignement de l'écoute musicale », thèse de doctorat, université Paris 6, 1990.

[94] S.Russell, P.Norvig. «Artificial Intelligence: A Modern Approach», The Intelligent Agent Book. Prentice Hall Series in Artificial Intelligence, 1995.

[95] S.Russell, P.Norvig. «Artificial Intelligence A Modern Approach», Prentice-Hall, 1995.

[96] B.Sadok, K.Ghédira. «OPT-D3G2A: a new dynamic distributed double guided genetic algorithm for Sigma-CSPs», Congress on Evolutionary Computation, 2005

[97] A.Saidane, H.Akdag, Isis Truck. « Une Approche SMA de l'Agrégation et de la Coopération des classifieurs », 3eme International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, Tunisia, 2005.

[98] B.Scherrer. « Apprentissage de représentation et auto-organisation modulaire pour un agent autonome », thèse doctorat, université Henri Poincaré – Nancy 1, 2003.

[99] S.Schulenburg, P.Ross. « An adaptive agent basel economic model », in P. L. Lanzi, W. Stoltzmann & S. W. Wilson, eds, "Learning classifier systems. From foundations to applications", Vol. 1813 of Lecture notes in artificial intelligence, Springer, Berlin, pp 263– 282. 2000.

[100] D.Schwab, M.Lafourcade, V.Prince. « Vers l'apprentissage automatique, pour et par les vecteurs conceptuels, de fonctions lexicales. L'exemple de l'antonymie ». TALN, Nancy,2002.

[101] M.Sébastien. « Apprentissage supervisé pour la cartographique », thèse de doctorat, université Pièrre et Marie Curie, Paris, 2001.

[102] S.A.Selouani, J.Caelen. « Un système hybride pour l'identification de traits phonétiques complexes de la langue arabe », NSIP : pp 709-713, 1999.

[103] S.SEN, G.Weiss. «Learning in Multiagent Systems» , WEISS G., Ed., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, 259-298,The MIT Press, Cambridge, MA, 1999.

[104] S.Sen. « Adaptation, coevolution and learning in multiagent systems», Technical report SS-96-01, Menlo Park, CA :AAAI Press, 1996.

[105] S.Singh, T.Jaakkola, M.L. Littman, C.Szepesvàri. «Convergence results for single-step on-policy reinforcement-learning algorithms». Machine Learning, 38(3):287–308, 2000.

[106] M.Slimane, Gilles Venturini, Jean Pierre Asselin de Beauville, Thierry Brouard, A. Brandeau. « Optimizing Hidden Markov Models with a Genetic Algorithm ». Artificial Evolution, Artificial Evolution, European Conference, AE 95, Brest, France, 1995.

[107] G.Stempfel, L.Ralaivola, F.Denis. « Apprentissage de SVM sur Données Bruitées », ALT,2007.

[108] N.Stroppa. « Définitions et caractérisations de modèles à base d'analogies pour l'apprentissage automatique des langues naturelles », thèse de doctorat, École Nationale Supérieure des Télécommunications, Paris, 2005.

[109] G.Syswerda. « Uniform crossover in genetic algorithms ». In Proceedings of the Third International Conference on Genetic Algorithms, pp 2-8. Morgan Kauffman. 1989.

[110] Tan. « Multi-agent reinforcement learning: independent vs cooperative agents». In ICML, pp 330–337, 1993.

[111] G. Tesauro, J. O. Kephart. « Pricing in agent economies using multi-agent Q-learning». Autonomous Agents and Multi-Agent Systems archive , 5(3) :289-304, 2002.

[112] M.Tien Tho DO. « Optimisation de forme en forgeage 3D », thèse de doctorat, École des mines de paris, 2006.

[113] J.Torres_Moreno. « Apprentissage et généralisation par des réseaux de neurones : étude de nouveaux algorithmes constructifs », thèse de doctorat, Institut National Polytechnique de Grenoble, France, 1997.

[114] T.Vallée, M.Yıldızoğlu. « Présentation des algorithmes génétiques et de leurs applications en économie », Revue d'économie politique, Vol. 114 : n° 6, 2004.

[115] T.Vallée. « Heterogeneous inflation learning : communication versus experiments ». Working Paper presenté à la conférence : Complex behavior in economics : modeling, computing, and mastering complexity, Aix en Provence, Marseille, France, 2000.

[116] B.Virole. « Réseaux de neurones et psychométrie », éditions du centre de psychologie Appliquée, 2001.

[117] N.Vriend. «An illustration of the essential difference between individual and social learning, and its consequences for computational analysis», Journal of Economic Dynamics and Control 24, pp 1–19. 2000

[118] G.Weiß . «Adaptation and Learning in Multi-Agent Systems». In Lecture Notes in artificial Intelligence 1042. Springer Verlag. 1996.

[119] G.Weiß. «Learning To Coordinate Actions In Multi-Agent Systems», in Proceedings of the International Joint Conference on Artificial Intelligence, pp 311-316, 1993.

[120] M.Yildizoglu. «Connecting adaptive behaviour and expectations in models of innovation: The potential role of artificial neural networks», European Journal of Economic and Social Systems, Vol. 15, n°2, pp 203-220. 2001.

[121] R.Zegadi. « Diagnostic des défauts par un couplage neurones artificiels - algorithmes génétiques », 4th International Conference on Computer Integrated Manufacturing CIP'2007, 2007.

Liste de publications

- A.Bendahmene, O.Kazar. «An Approach Based on Genetic Algorithm for the Learning of an Agent», Information Systems and Econimic Intelligence, 4th International Conference, SIIE'2011, pp 308-314, Marrakech, Marroc. Febrary 17th-19th, 2011.
- Un article correspondant à mon travail, est accepté à la conférence International Conference on Information Science and Technology, ICIST'2011, Tebessa, Algérie. Il sera publié dans le proceeding et présenté le 24-26 avril 2011.
- Un article correspondant à mon travail, est accepté à la conférence: The 2nd International Conference on Information and Communication Systems, ICICS'2011, Aman, Jordon. Il sera publié dans le proceeding et présenté le Mai 22-24, 2011.