



MEMOIRE

Présenté par

Belghoul Riyane

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Modélisation de l'adversaire dans un domaine de
négociation multi issus non linéaire

Soutenu le : 25 / 06 / 2023

Devant le Jury composé de :

Président	Mr. Mme. XXX	MCB	Chadli Bendjedid El-Tarf
Rapporteur	Mr. Betouil Ali Abdelatif	MCB	Chadli Bendjedid El-Tarf
Examineur	Mr. Mme. ZZZ	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2022/2023

Remerciements

*Tout d'abord je tiens à remercier **ALLAH** qui m'a donné la force l'intelligence et la patience d'accomplir ce modeste travail.*

*Mes remerciements et ma gratitude s'adressent à mon encadreur **Dr Betouil Ali Abdelatif** pour sa disponibilité et l'intérêt qu'il a porté à mon travail tout au long de semestre.*

*Je souhaite remercier **Dr** .pour avoir accepté de présider le jury, qu'elle me permet de lui exprimer ma plus haute considération.*

*Je remercie également **Dr** .pour leur investissement en tant qu'examineur et pour le temps qu'il a consacré à lire et évaluer ce travail.*

Un grand merci ainsi que tous les enseignants de département Mathématique et informatique.

Enfin, je remercie mes chers parents, ma famille et mes amis qui m'ont encouragée tout au long de mes études.

Leur soutien et leur aide sont pour beaucoup dans l'accomplissement de ce travail.

Du profond dans mon cœur, je dédie ce travail à tous ce qui est chères.

A mes chers parents

Que ce modeste travail soit l'exaucement de vos vœux tout formulés, le fruit de vos innombrables sacrifices. Puisse Dieu le très Haut, vous accorder santé, bonheur et longue vie.

Mes sœurs et ma famille

C'est un moment de plaisir de dédier ce travail, en signe d'amour, de reconnaissance et de gratitude pour le dévouement dont vous avez fait toujours preuve à mon égard.

A mes meilleurs amis

merci pour être dans ma vie je vous aime de fond du mon cœur.

La négociation est un processus de communication et de prise de décision qui vise à résoudre des divergences d'intérêts entre différentes parties impliquées.

Dans ce travail, nous étudions les mécanismes de négociation bilatérale multi-issues dans un domaine non linéaire dans le but d'atteindre une solution de compromis satisfaisante l'objectif est de développer une approche qui permet de modéliser l'adversaire en se basant sur les négociations précédentes.

Dans cette étude, nous concevons un agent utilisant l'apprentissage par renforcement et mis en œuvre avec l'algorithme Q-learning pour anticiper les intentions d'autres agents et les actions futures envisageables.

Notre objectif est de maximiser le rôle de notre agent en prenant des décisions concernant la sélection de la meilleure offre parmi plusieurs offres multi-issues non linéaires, en tenant compte des stratégies d'acceptation/refus, de concessions et de proposition. Nous démontrons que notre agent est capable d'apprendre dans un environnement multi-issues non linéaire et d'obtenir une meilleure valeur d'utilité que les autres agents disponibles dans la plateforme de simulation des négociations automatiques appelée GENIUS.

Les mots clés : négociation automatique, agent, GENIUS , apprentissage par renforcement, multi issues non-linéaire.

Negotiation is a communication and decision-making process that aims to resolve conflicting interests between different parties involved.

In this work, we study the mechanisms of bilateral multi-issue negotiation in a nonlinear domain with the aim of reaching a satisfactory compromise solution. The objective is to develop an approach which makes it possible to model the adversary based on the previous negotiations.

In this study, we design an agent using reinforcement learning and implemented with the Q-learning algorithm to anticipate the intentions of other agents and possible future actions.

Our goal is to maximize the role of our agent in making decisions regarding the selection of the best offer among several non-linear multi-issue offers, taking into account accept/decline, concessions and proposal strategies. We demonstrate that our agent is able to learn in a nonlinear multi-issue environment and to obtain a better utility value than the other agents available in the automatic trading simulation platform called GENIUS.

Keywords: automatic negotiation, agent, GENIUS, reinforcement learning, non-linear multi-issues.

التفاوض هو عملية اتصال واتخاذ قرار تهدف إلى حل المصالح المتضاربة بين مختلف الأطراف المعنية .
في هذا العمل ، ندرس آليات التفاوض الثنائي متعدد القضايا في مجال غير خطي بهدف الوصول إلى حل وسط مرضٍ ،
والهدف هو تطوير نهج يجعل من الممكن نمذجة الخصم بناءً على المفاوضات السابقة.
في هذه الدراسة ، قمنا بتصميم وكيل باستخدام التعلم المعزز وتنفيذها باستخدام خوارزمية Q-Learning لتوقع نوايا الوكلاء
الأخرين والإجراءات المستقبلية المحتملة.
هدفنا هو تعظيم دور وكيلنا في اتخاذ القرارات المتعلقة باختيار أفضل عرض من بين عدة عروض غير خطية متعددة
القضايا ، مع مراعاة استراتيجيات القبول / الرفض والامتيازات والاقتراح. نثبت أن وكيلنا قادر على التعلم في بيئة غير
خطية متعددة القضايا والحصول على قيمة فائدة أفضل من الوكلاء الآخرين المتاحين في منصة محاكاة التداول التلقائية
GENIUS المسماة

الكلمات المفتاحية: التفاوض الآلي ، الوكيل ، العبقورية ، التعلم المعزز ، القضايا المتعددة غير الخطية.

Chapitre 1 : les systèmes multi agents	19
1 Introduction :	20
2 Les approches de négociation :	20
2.1 La négociation compétitive :	20
2.1.1 Caractéristiques de négociations de compétitive:	20
2.2 La négociation collaborative :	21
2.2.1 Caractéristiques de négociations de collaborative:	21
2.3 La négociation raisonnée:	21
2.3.1 Caractéristiques de négociations de raisonnée:	22
2.4 La négociation basée sur les intérêts :	22
2.4.1 Caractéristiques de négociation basée sur les intérêts:	23
3 utilisation d'IA pour résoudre un problème :	23
3.1 Définir clairement le problème:	23
3.2 Collecter les données:	24
3.3 Choisir la technique appropriée:	24
3.4 Entraîner le modèle:	24
3.5 Évaluer le modèle:	24
3.6 Déployer le modèle:	24
4 l'avantage d'utilisation du système multi agent :	24
4.1 Système dynamique:	24
4.2 Distribution du traitement :	25
4.3 Adaptabilité :	25
4.4 Résilience :	25
4.5 Scalabilité :	25
4.6 Modélisation de la complexité :	25
4.7 Adaptabilité à la diversité :	26
5 la négociation dans SMA :	26

6	Les techniques de négociation des agents :	26
6.1	La négociation bilatérale :	27
6.2	La négociation multilatérale :	27
6.3	La négociation mini latérale :	27
7	Les avantages et les inconvénients de chaque technique de négociation :	27
8	Comparaison entre les techniques de négociation :	29
8.1	Négociation bilatérale :	29
8.2	Négociation multilatérale :	29
8.3	Négociation mini-latérale :	30
9	La négociation multi-issue :	30
10	conclusion.....	31
	Chapitre 2 : Machine Learning	32
1	Introduction :	33
2	Définition de Machine Learning:	33
3	L'objectif de Machine Learning :	33
4	Les algorithmes de Machine Learning :	34
4.1	l'apprentissage supervisé :	34
4.1.1	définition :	34
4.1.2	Les applications de l'apprentissage supervisé :	35
4.1.3	les algorithmes d'apprentissage supervisé :	35
4.1.3.1	Régression linéaire :	35
4.1.3.2	la régression logistique :	36
4.1.3.3	SVM :	37
4.2	l'apprentissage non supervisé :	39
4.2.1	définition :	39
4.2.2	Les applications de l'apprentissage non supervisé :	39
4.2.3	Les algorithmes de l'apprentissage non supervisé :	39
4.2.3.1	K means :	39
4.2.3.2	Algorithme Apriori :	40
4.2.3.3	clustering hiérarchique :	41

4.3	l'apprentissage semi supervisé :	41
4.3.1	définition :	41
4.3.2	les applications d'apprentissage semi supervisé :	41
4.3.3	Les algorithmes de l'apprentissage semi supervisé :	42
4.3.3.1	Label Propagation Algorithme :	42
4.3.3.2	L'apprentissage par co-enseignement (co-training) :	42
4.4	l'apprentissage par renforcement :	42
4.5	Deep learning :	43
4.6	Réseau de neurones artificiels (ANN) :	44
4.6.1	Les réseaux de neurones convolutifs (CNN) :	44
4.6.2	Réseau neuronal récurrent (RNN) :	45
5	Conclusion :	46
Chapitre 3 : Les travaux de ML dans la négociation de SMA		47
1	Introduction :	48
2	Les travaux de Machine Learning :	48
2.1	Apprentissage par renforcement multi-agents pour la négociation avec des adversaires inconnus :	48
2.2	Apprendre à négocier dans des systèmes multi-agents à l'aide de l'apprentissage par renforcement profond :	49
2.3	Apprentissage par renforcement profond pour la négociation (Learning from Experience and Dialogue Feedback) :	49
2.4	Conception de mécanismes d'enchères avec apprentissage par renforcement dans des environnements multi-agents :	50
2.5	Apprentissage bayésien :	51
2.6	Estimation de la densité du noyau :	51
2.7	Régression non _ linéaire :	51
2.8	Enquête sur l'apprentissage dans les environnements multi-agents (Gérer la non-stationnarité) :	52
3	Algorithme Q-Learning :	53
3.1	Définition :	53
3.2	Le Q-table :	54

3.3	Le principe :	55
4	Conclusion :	56
Chapitre 4 : conception de notre agent		57
1	Introduction :	58
2	Description de notre agent :	58
3	Conception de notre agent :	58
4	les stratégies de négociation :	59
4.1	Stratégie d'apprentissage :	60
4.1.1	l'étape de mémorisation :	60
4.1.2	L'étape de renforcement :	60
4.1.3	L'étape d'évaluation :	60
4.1.4	L'étape de décision :	60
4.2	Stratégie d'acceptation :	60
4.3	Stratégie de proposition :	61
4.4	La stratégie de concession :	62
5	Conclusion :	62
Chapitre 5 : Implémentation et résultats		63
1	Introduction :	64
2	La plateforme GENIUS :	64
2.1	Présentation des interfaces du GENIUS :	64
3	La compétition des agents de négociation automatisés ANAC :	68
4	Eclipse :	68
5	java (JDK) :	68
6	Discussion des résultants :	69
6.1	Exemple d'application dans une session de négociation :	69
6.2	Résultats obtenus :	71
7	Conclusion :	74

Listes des figures

Figure 1 : La régression linéaire	35
Figure 2: La régression logistique	36
Figure 3: Machine à vecteurs de support	37
Figure 4: un espace vectoriel de plus grande dimension	38
Figure 5: LPA	42
Figure 6: Schéma d'interaction entre l'agent et son environnement.....	43
Figure 7 : un réseau de neurones artificiels	44
Figure 8: réseaux de neurones convolutifs	45
Figure 9: les couches de réseau de neurones	45
Figure 10 : Réseau de neurones récurrents	46
Figure 11: Un mécanisme d'apprentissage par renforcement pour une enchère conception	50
Figure 12: Explication de la fonction de Q-learning.....	54
Figure 13: . Le Q-table	54
Figure 14: algorithme Q-Learning	55
Figure 15: Stratégie de négociation.....	60
Figure 16: La stratégie d'acceptation	61
Figure 17: la stratégie de proposition	61
Figure 18: stratégie de concession	62
Figure 19: GENIUS juste après le démarrage, à la gauche on a le panneau de composants à la droite Le panneau d'état.	65
Figure 20: Les agents intégrés avec GENIUS.	65
Figure 21: : composants BOA ; offrant à l'utilisateur la possibilité de créer et d'appliquer des composants nouvellement développés utiliser une interface utilisateur graphique.	66
Figure 22: Création d'une négociation session simple.	67
Figure 23 : Le résultat de la négociation.	67
Figure 24: : Choisir les paramètres d'une négociation, ainsi que le protocole de négociation, deadline et les agents et leurs préférences, enfin, cliqué « start » pour commencer la négociation.....	67
Figure 25: La session de négociation est selon le protocole alternative, notre agent.....	69
Figure 26: Les résultats selon la négociation session précédente. Nous remarquons	70
Figure 27: Une session tournoi avec tous les agents compatibles avec les utilités nonlinéaires sous GENIUS.....	72

Figure 28: Les résultats du tournoi ; La boîte à outils donne de précieux graphiques	73
Figure 29: Tableau de résultats	73
Figure 30: Suite tableau de résultats.....	74

Liste des tableaux

Tableau 1: Les avantages et les inconvénients de chaque technique de négociation.....29

Liste des équations

Équation 1: équation de Bellman	53
Équation 2: équation de Q-Learning	56

- α**Coefficient d'apprentissage.
- γ**Coefficient d'actualisation.
- R**.....Fonction de récompense.
- r**.....Récompense scalaire.
- T**.....Deadline.
- t**.....Temps Période d'échantillonnage.
- A**.....Ensemble d'action.
- a**.....Action.
- S**.....Ensemble des états.
- s**.....Etat courant.
- Q**.....Fonction d'utilité.
- B**.....Une valeur réservé de l'utilité minimale.
- $U_{A(W^{t_{B \rightarrow A}})}$L'utilité de l'agent «A» qui lui gagner d'après la proposition de « B ».
- $U_{A(W^{t_{A \rightarrow B}})}$L'utilité de l'agent «A» qui lui gagner d'après leur proposition chez l'agent « B ».

La négociation est un processus où toutes les parties s'engagent à trouver une solution. Elles commencent par exprimer leurs exigences respectives et font des compromis successifs pour parvenir à un accord ou explorer de nouvelles options. Les négociations se distinguent en deux catégories : la négociation compétitive et la négociation coopérative. Dans les négociations compétitives, les agents sont en concurrence les uns avec les autres pour maximiser leur utilité individuelle. En revanche, dans les négociations coopératives, les agents cherchent à atteindre une efficacité globale maximale, en prenant en compte la valeur de toutes leurs activités. Ce type de négociation diffère considérablement de la négociation compétitive et peut être considéré comme un processus de recherche décentralisé.

La négociation que nous abordons dans ce mémoire est de nature compétitive, car notre objectif est de maximiser l'utilité de notre agent négociateur.

Ce mémoire porte sur les modèles décisionnels des agents dans le contexte de négociations multi-issues et non linéaires. L'objectif est de fournir une stratégie d'apprentissage permettant à un agent de prévoir les intentions et les actions futures possibles d'un autre agent adverse afin de maximiser son utilité locale et de remporter la négociation. Les modèles utilisés par les agents sont non linéaires car ils doivent établir leur propre représentation à travers de multiples interactions avec l'environnement.

Nous abordons la création de stratégies d'acceptation, où l'agent décide des conditions pour maximiser son utilité, ainsi que la concession, où l'agent renonce à certaines demandes pour son bénéfice, et la proposition, où l'agent refuse une offre adverse et propose une contre-offre pour augmenter son utilité.

Notre défi dans ce travail est de : modélisation de l'adversaire dans un domaine de négociation multi-issues non linéaire en utilisant une modélisation et un algorithme basé sur l'apprentissage par renforcement.

Pour atteindre cet objectif, nous développons un agent capable de remporter une négociation en maximisant la valeur d'utilité grâce à l'utilisation de fonctions d'apprentissage par renforcement et de l'algorithme Q-learning.

Ce mémoire est organisé en cinq chapitres :

- ✓ Dans le premier chapitre : ce chapitre est composé de deux parties, la première partie nous présentons les approches de négociation, l'utilisation de IA pour résoudre un

problème , l'avantage d'utilisation des système multi agent , la négociation dans système multi agent , les techniques de négociation ,les avantages et les inconvénients de chaque techniques, comparaison entre les techniques de négociation , Dans Le deuxième nous fournissons une définition sur la négociation multi issues.

- ✓ Dans le deuxième chapitre : nous présentons le Machine Learning et le fonctionnement des algorithmes d'apprentissage automatique les plus populaires.
- ✓ Dans le troisième chapitre : nous appuyons les travaux existants qui utilisent ML dans la négociation de SMA , l'algorithme de l'apprentissage le Q-Learning.
- ✓ Dans le quatrième chapitre : nous présentons une brève présentation de notre agent négociateur, puis nous aborderons l'approche d'apprentissage par renforcement.
- ✓ Dans le cinquième chapitre : Nous présentons la partie expérimentale de notre travail et analysons les résultats obtenus.

Chapitre 1 : les systèmes multi agents

1 Introduction :

Les systèmes multi-agents sont un sous-domaine de l'intelligence artificielle distribuée qui a connu une croissance rapide en raison de la flexibilité et de l'intelligence disponibles pour résoudre les problèmes distribués [1]. et la négociation est une activité courante dans les systèmes multi-agents où les agents doivent coopérer ou rivaliser pour atteindre leurs objectifs. La négociation peut être définie comme un processus d'échange de propositions entre des agents autonomes, qui visent à parvenir à un accord satisfaisant pour les deux parties. Dans ce chapitre bref en va voir des différents attributs tels que les approches de négociation , l'utilisation de IA pour résoudre un problème , l'avantage d'utilisation des système multi agent , la négociation dans système multi agent , les techniques de négociation , les avantages et les inconvénients de chaque techniques, comparaison entre les techniques de négociation , négociation multi issus .

2 Les approches de négociation :

Les approches de négociation entre les agents peuvent varier en fonction de nombreux facteurs tels que la culture, les objectifs et les relations entre les parties impliquées. Cependant, voici quelques-unes des approches de négociation les plus courantes :

2.1 La négociation compétitive :

Cette approche est également connue sous le nom de « gagnant-perdant ». Les parties impliquées cherchent à maximiser leurs propres gains et à minimiser les gains de l'autre partie. Cela peut entraîner des conflits et des situations où une partie sort perdante.

2.1.1 Caractéristiques de négociations de compétitive:

Les principales caractéristiques de négociations de compétitive ou de distribution comprennent :

- est un type gagnant-perdant.
- Une seule solution à un problème est négociée, s'il y en a plusieurs, la négociation d'intégration.
- Il existe une corrélation négative dans la réalisation des objectifs.
- Il n'est pas rare que ce type de négociation débouche sur une situation fermée.

- Les intérêts sont contradictoires et une confrontation s'ensuit.
- Ils ont tendance à créer des modèles de supériorité-infériorité parmi les participants, déformant les perceptions de la relation.
- est conçu comme une relation destructrice.
- Il s'agit d'une relation à court terme. [2]

2.2 La négociation collaborative :

Cette approche implique la coopération entre les parties impliquées pour atteindre un objectif commun. Les parties cherchent à travailler ensemble pour trouver une solution qui profite à tous les membres de l'équipe. La collaboration est comprise comme une coopération améliorée par le développement d'une compréhension mutuelle par rapport à la vision commune de la tâche à résoudre par différentes entités en interaction [3] [4]. ajoute que la coopération est une coopération pour laquelle les acteurs ont un objectif commun et aucun autre objectif personnel incompatible.

2.2.1 Caractéristiques de négociations de collaborative:

- La négociation collaborative est la divulgation spontanée d'informations. [5]
- L'accord sur les négociations conjointes est en principe juridiquement contraignant (règlement) [6]
- identifie les parties, définit le conflit, inclut le principe de confidentialité, facultatif. [7]
- Les différends seront résolus d'une manière jugée équitable par les parties. [8]
- Les parties s'efforcent d'atteindre un résultat final mutuellement satisfaisant et surtout ne craignent pas les ténèbres de l'avenir. [9]

2.3 La négociation raisonnée:

cette approche se concentre sur les principes qui sous-tendent la négociation plutôt que sur les positions individuelles des parties impliquées. Les parties cherchent à trouver une solution qui respecte les principes tels que la justice, l'équité et la dignité. Ce processus est particulièrement utile dans les négociations complexes ou les conflits émotionnels, car il aide les parties à travailler ensemble pour trouver des solutions qui répondent à leurs besoins mutuels de manière efficace et respectueuse.

2.3.1 Caractéristiques de négociations de raisonnée:

- Focalisation sur les principes: Les négociations raisonnées se concentrent sur des principes et des normes qui ont de l'importance pour toutes les parties. Les principes peuvent inclure des normes morales, des règles, des lois ou des standards de qualité.
- Communication ouverte: Les négociations raisonnées impliquent une communication ouverte et honnête entre les parties. Les parties sont transparentes sur leurs intérêts, leurs besoins et leurs préoccupations, ce qui facilite la création de solutions mutuellement satisfaisantes.
- Exploration de solutions multiples: Dans les négociations raisonnées, les parties cherchent à trouver des solutions créatives qui répondent aux besoins des deux parties. Cela peut inclure la création de solutions innovantes qui n'ont pas été envisagées auparavant.
- La créativité : les négociateurs raisonnés cherchent des solutions innovantes en explorant toutes les options possibles pour répondre aux besoins de chaque partie.
- La compréhension mutuelle : les négociateurs raisonnés cherchent à comprendre les perspectives et les intérêts de chaque partie afin de trouver des solutions qui répondent aux besoins de tous.
- Recherche de gains mutuels: Dans les négociations raisonnées, les parties cherchent des solutions qui bénéficient aux deux parties. Les parties travaillent ensemble pour trouver des solutions qui créent de la valeur pour tous plutôt que de simplement diviser un gâteau fixe.

2.4 La négociation basée sur les intérêts :

cette approche se concentre sur les intérêts des parties impliquées plutôt que sur leurs positions. Les parties cherchent à comprendre les motivations de l'autre partie et à trouver une solution qui satisfait les intérêts de toutes les parties. La négociation basée sur les intérêts est souvent considérée comme plus efficace que les approches de négociation plus traditionnelles, telles que la négociation axée sur les positions, car elle permet de trouver des solutions plus durables et plus satisfaisantes pour toutes les parties impliquées. Elle nécessite cependant une forte communication et une écoute active de la part de chaque partie pour identifier et comprendre les intérêts de l'autre.

2.4.1 Caractéristiques de négociation basée sur les intérêts:

- Voici quelques caractéristiques de négociations basées sur les intérêts :
- Focus sur les intérêts: Les négociations basées sur les intérêts mettent l'accent sur les intérêts et les besoins réels des parties impliquées plutôt que sur leurs positions. Les parties cherchent à comprendre ce qui est important pour l'autre partie afin de trouver une solution mutuellement satisfaisante.
- Collaboration plutôt que compétition: Les négociations basées sur les intérêts encouragent la collaboration entre les parties plutôt que la compétition. Les parties cherchent à trouver des solutions gagnant-gagnant, où chacun peut obtenir ce qu'il veut.
- Exploration de solutions multiples: Dans les négociations basées sur les intérêts, les parties cherchent à trouver des solutions créatives qui répondent aux besoins des deux parties. Cela peut inclure la création de solutions innovantes qui n'ont pas été envisagées auparavant.
- Communication ouverte: Les négociations basées sur les intérêts impliquent une communication ouverte et honnête entre les parties. Les parties sont transparentes sur leurs intérêts, leurs besoins et leurs préoccupations, ce qui facilite la création de solutions mutuellement satisfaisantes.
- Flexibilité: Les négociations basées sur les intérêts sont souvent flexibles et adaptables. Les parties sont disposées à modifier leur position ou à considérer d'autres options si cela peut aider à atteindre un accord satisfaisant pour les deux parties.
- Recherche de gains mutuels: Dans les négociations basées sur les intérêts, les parties cherchent des solutions qui bénéficient aux deux parties. Les parties travaillent ensemble pour trouver des solutions qui créent de la valeur pour tous plutôt que de simplement diviser un gâteau fixe.

3 utilisation d'IA pour résoudre un problème :

Pour utiliser l'IA pour résoudre un problème, il est important de suivre certaines étapes clés:

3.1 Définir clairement le problème:

avant de commencer à utiliser l'IA pour résoudre un problème, il est important de bien comprendre le problème et d'identifier les données pertinentes qui peuvent être utilisées pour résoudre le problème.

3.2 Collecter les données:

ne fois que vous avez identifié les données pertinentes, vous devez collecter et nettoyer ces données afin qu'elles soient utilisables pour les algorithmes d'IA.

3.3 Choisir la technique appropriée:

Il existe plusieurs techniques d'IA, notamment l'apprentissage automatique, le traitement du langage naturel, la vision par ordinateur, la robotique, etc. Il est important de choisir la technique appropriée en fonction du type de problème à résoudre.

3.4 Entraîner le modèle:

une fois que vous avez choisi la technique appropriée, vous devez entraîner le modèle à l'aide des données collectées. Cela peut nécessiter beaucoup de temps et de ressources informatiques, mais c'est une étape critique pour que le modèle puisse apprendre à résoudre le problème.

3.5 Évaluer le modèle:

après avoir entraîné le modèle, vous devez l'évaluer pour voir s'il peut résoudre le problème de manière efficace. Si le modèle n'est pas satisfaisant, vous devrez peut-être ajuster les paramètres ou ajouter plus de données pour améliorer les performances.

3.6 Déployer le modèle:

une fois que vous êtes satisfait des performances du modèle, vous pouvez le déployer pour résoudre le problème. Cela peut nécessiter l'intégration du modèle dans un système existant ou la création d'une application dédiée pour résoudre le problème.

4 l'avantage d'utilisation du système multi agent :

Les systèmes multi-agents (SMA) présentent de nombreux avantages par rapport aux systèmes monolithiques ou centralisés. Voici quelques-uns des avantages clés :

4.1 Système dynamique:

Les systèmes multi-agents (SMA) bénéficient des avancées de l'IA en termes de traitement symbolique et de gestion des connaissances. Contrairement aux approches classiques de l'IA qui simulent partiellement les capacités du comportement humain, les SMA permettent de modéliser un ensemble d'agents qui interagissent entre eux. [10]

4.2 Distribution du traitement :

Les SMA permettent de répartir le traitement sur plusieurs agents, ce qui permet de traiter des problèmes complexes de manière plus efficace et plus rapide. Chaque agent peut travailler de manière autonome en effectuant des tâches spécifiques qui contribuent à la réalisation de l'objectif global. L'utilisation d'agents spécialisés peut améliorer l'efficacité du traitement en permettant aux agents de se concentrer sur leurs tâches spécifiques, Les agents peuvent également travailler en collaboration, en partageant des informations et en coordonnant leurs actions pour atteindre des objectifs communs.

4.3 Adaptabilité :

Les SMA sont conçus pour s'adapter à des environnements dynamiques et incertains. Les agents peuvent utiliser des mécanismes d'apprentissage et de prise de décision distribuée pour ajuster leur comportement en fonction des changements dans leur environnement ou des interactions avec d'autres agents. Les agents peuvent également prendre des décisions en temps réel, ce qui leur permet de s'adapter rapidement aux situations changeantes.

4.4 Résilience :

Les SMA sont décentralisés, ce qui signifie qu'ils sont résilients aux pannes et aux défaillances. Si un agent échoue, les autres agents peuvent continuer à travailler sans interruption. Les SMA sont également capables de se réorganiser pour maintenir le fonctionnement global du système en cas de panne ou de défaillance.

4.5 Scalabilité :

Les SMA peuvent être facilement étendus en ajoutant de nouveaux agents, ce qui permet de résoudre des problèmes de plus grande envergure sans augmenter la complexité du système dans son ensemble. Les SMA peuvent également être utilisés pour résoudre des problèmes à grande échelle en distribuant la charge de travail sur plusieurs agents.

4.6 Modélisation de la complexité :

Les SMA peuvent être utilisés pour modéliser des systèmes complexes avec de nombreux acteurs et interactions. Les agents peuvent représenter des personnes, des organisations, des machines ou d'autres entités, et les interactions entre les agents peuvent être utilisées pour simuler le comportement du système. Les SMA sont particulièrement utiles pour résoudre des problèmes où les interactions entre les acteurs sont complexes et difficiles à modéliser.

4.7 Adaptabilité à la diversité :

Les SMA sont capables de gérer des situations diverses en utilisant des agents spécialisés. Les agents peuvent avoir des compétences différentes, des connaissances et des préférences qui peuvent être utilisées pour résoudre des problèmes spécifiques. Les SMA peuvent également être utilisés pour résoudre des problèmes dans des domaines variés, tels que la finance, l'industrie, les sciences de la vie, etc.

5 la négociation dans SMA :

Négociation est une interaction entre agents basée sur communication pour parvenir à un accord [11] .

En vue de modéliser la négociation dans un SMA, il est habituel de considérer les éléments suivants :

- ✚ Le langage de négociation : Il s'agit d'un ensemble de primitives de communication qui décrit comment les agents communiquent entre eux.
- ✚ Le protocole de négociation : Il désigne l'ensemble des règles régissant la négociation, telles que les propositions licites ou les états de la négociation.
- ✚ Les objets de négociation : Il s'agit des éléments sur lesquels porte la négociation, tels que le prix, les délais ou les temps de réponse.
- ✚ Le processus de décision : C'est le cœur du fonctionnement et de l'intelligence de la négociation, car il oriente la stratégie de l'agent au cours de la négociation. La stratégie peut être prédéfinie et immuable ou bien modifiée en fonction des stratégies des autres agents (observation, supposition).

Il convient également de tenir compte de la cardinalité des participants à la négociation, qui peut varier entre une négociation un-à-un, une négociation un-à-plusieurs ou une négociation plusieurs-à-plusieurs. [12]

6 Les techniques de négociation des agents :

Les techniques de négociation des agents sont des règles et des procédures qui permettent à des agents autonomes de communiquer et de négocier entre eux afin d'atteindre des objectifs communs ou individuels. Les agents autonomes peuvent être des robots, des programmes informatiques ou des entités virtuelles qui ont la capacité de prendre des décisions et d'agir de

manière autonome. Il existe différents types de techniques de négociation des agents c'est comme suit :

6.1 La négociation bilatérale :

La négociation bilatérale est un type de négociation qui implique deux parties qui cherchent à parvenir à un accord ou à une entente. Ce type de négociation peut se produire dans de nombreux contextes, tels que les négociations commerciales, les négociations de contrats, les négociations salariales, les négociations diplomatiques, ou encore les négociations de divorce.

6.2 La négociation multilatérale :

La négociation multilatérale est un processus impliquant plusieurs parties des échanges qui permettent aux participants ayant des intérêts différents d'atteindre des objectifs ensemble, Ces participants ont le comportement égoïste le plus courant, et ne partagent pas d'informations à leur sujet fonctions utilitaires et leurs préférences. chaque essayer d'orienter les négociations vers un accord' atteindre ses objectifs. dans une telle circonstance, Lorsque la quantité Les participants comptent. [13] La plupart des études menées sur la négociation multilatérale ne proposent pas de protocoles spécifiques ou reposent sur des hypothèses difficilement applicables dans un contexte réel. Certains auteurs étendent les mécanismes de négociation développés pour le cas d'une négociation bilatérale [14]. Par exemple, propose une généralisation du protocole de concession monotone [15] pour la négociation multilatérale, mais ne fournit pas de protocole spécifique.

6.3 La négociation mini latérale :

la négociation mini latérale est un processus de négociation impliquant un petit nombre de parties ayant des intérêts communs, et peut être utilisée pour résoudre des problèmes régionaux ou spécifiques, ou pour faire avancer des questions dans des domaines tels que le commerce, l'environnement ou la sécurité. et peuvent impliquer des rencontres informelles ou des pourparlers en petit comité. Les négociateurs peuvent également avoir plus de flexibilité pour explorer des options créatives ou innovantes pour résoudre les problèmes.

7 Les avantages et les inconvénients de chaque technique de négociation :

Technique de négociation	Les avantages	Les inconvénients
Négociation bilatérale	<ul style="list-style-type: none"> - Rapidité et efficacité dans la prise de décisions - Meilleure capacité à contrôler le processus de négociation et à défendre les intérêts de chaque partie. - Plus grande flexibilité pour s'adapter aux besoins et aux intérêts de chaque partie. - Plus de confidentialité, car les négociations se déroulent en petit comité. 	<ul style="list-style-type: none"> - Risque de déséquilibre des pouvoirs entre les deux parties, ce qui peut affecter le résultat de la négociation. - Moins d'opportunités pour trouver des solutions créatives et innovantes, car il y a moins de parties impliquées. - Risque de conflits et de désaccords si les parties ont des positions diamétralement opposées.
Négociation multilatérale	<ul style="list-style-type: none"> - Meilleure représentation des différents intérêts et perspectives des parties prenantes - Plus grande opportunité de trouver des solutions créatives et innovantes, car il y a plus de parties impliquées. - Plus grande légitimité, car les résultats sont soutenus par un plus grand nombre de parties prenantes. - Moins de risques de conflits et de désaccords, car les parties prennent en compte les intérêts des autres parties. 	<ul style="list-style-type: none"> - Risque de blocage si les parties ne parviennent pas à trouver un terrain d'entente. - Plus de difficultés à s'adapter aux besoins et aux intérêts de chaque partie, car il y a plus de parties impliquées. - Plus de difficultés à s'adapter aux besoins et aux intérêts de chaque partie, car il y a plus de parties impliquées.

Négociation minilatérale	<ul style="list-style-type: none"> - Plus grande opportunité de trouver des solutions créatives et innovantes, car les parties partagent des intérêts et des préoccupations similaires. -Processus plus rapide et plus flexible que les négociations multilatérales. -Plus grande capacité à travailler ensemble en étroite collaboration 	<ul style="list-style-type: none"> - Risque de représentativité limitée si certaines parties prenantes sont exclues des négociations. -Risque de déséquilibre des pouvoirs si une partie est plus dominante que les autres. -Risque de ne pas prendre en compte les intérêts des parties prenantes qui ne sont pas incluses dans le processus de négociation.
--------------------------	--	--

Tableau 1: Les avantages et les inconvénients de chaque technique de négociation

8 Comparaison entre les techniques de négociation :

La négociation peut prendre plusieurs formes en fonction du nombre de parties impliquées et de la nature des accords qui en résultent. Les trois types de négociations les plus courants sont la négociation bilatérale, la négociation multilatérale et la négociation mini-latérale.

8.1 Négociation bilatérale :

- Deux parties impliquées
- Utilisée pour résoudre des problèmes spécifiques ou établir des relations commerciales
- Plus rapide et plus flexible que les négociations multilatérales
- Moins de parties impliquées
- Les parties peuvent se concentrer sur leurs intérêts respectifs
- Peut-être plus informelle et moins structurée

8.2 Négociation multilatérale :

- Trois ou plus de parties impliquées
- Utilisée pour négocier des accords commerciaux, des traités internationaux ou des accords environnementaux

- Plus complexe que les négociations bilatérales
- Prend plus de temps pour aboutir à un accord
- Tous les participants doivent prendre en compte les intérêts de toutes les parties
- Nécessite souvent des structures et des procédures formelles

8.3 Négociation mini-latérale :

- Petit groupe de parties impliquées
- Utilisée pour des questions spécifiques nécessitant l'expertise d'un petit groupe de parties
- Plus rapide que les négociations multilatérales
- Plus complexe que les négociations bilatérales
- Les parties doivent travailler ensemble pour parvenir à un accord qui convient à tous.
- Peut impliquer des discussions plus techniques et spécialisées
- Nécessite souvent une expertise spécifique

9 La négociation multi-issue :

la négociation multi-issues est un processus qui consiste à regrouper tous les enjeux et à en discuter simultanément. [16] Cette approche réduit le coût de transaction pour chaque accord de problème individuel, mais cela peut prendre beaucoup de ressources pour parvenir à un accord sur tous les problèmes simultanément, car il peut être difficile de diriger un processus de recherche dans un espace vaste et complexe. [17]

les parties impliquées cherchent à résoudre plusieurs problèmes ou questions en même temps. Contrairement à la négociation à issue unique, où les parties cherchent à parvenir à un seul accord sur une question spécifique, la négociation multi-issues implique souvent plusieurs problèmes à résoudre collectivement.

Les problèmes peuvent également être de nature différente, tels que les questions financières, les questions de personnel, les aspects techniques, les questions réglementaires, les questions juridiques, etc. Par exemple, les problèmes financiers peuvent inclure des problèmes de

liquidité, de rentabilité ou de gestion de la dette, tandis que les problèmes de personnel peuvent inclure des problèmes de rétention, de formation ou de motivation. Dans une négociation multi-issus, chaque partie peut avoir des priorités différentes et des objectifs distincts pour chaque question à résoudre.

Les objectifs de chaque partie peuvent également varier en fonction de la nature des problèmes. La négociation multi-issus est souvent utilisée dans des situations complexes où il y a plusieurs parties impliquées, des intérêts différents et des enjeux multiples. Les négociateurs doivent être en mesure de hiérarchiser les problèmes, de rechercher des solutions créatives et d'établir des compromis pour parvenir à un accord global qui convient à toutes les parties.

En fin, La négociation multi issus est un processus itératif qui nécessite une communication efficace et un engagement à long terme pour maintenir les accords.

Les parties doivent être disposées à travailler ensemble pour maintenir les relations et résoudre les problèmes à l'avenir.

10 conclusion

Dans ce chapitre nous avons vu les systèmes multi-agents et la négociation multi issus, les SMA sont des systèmes composés de plusieurs agents autonomes qui interagissent pour atteindre des objectifs communs, tandis que la négociation est un processus d'échange d'informations et de propositions entre deux ou plusieurs parties en vue d'arriver à un accord mutuellement satisfaisant. Les agents dans les systèmes multi-agents peuvent être programmés pour représenter des parties négociantes, et peuvent utiliser des techniques d'intelligence artificielle pour identifier les intérêts et les préférences des parties, proposer des offres et négocier pour parvenir à un accord. Et dans le chapitre suivant, nous verrons Machine Learning et les algorithmes de ML.

Chapitre 2 : Machine Learning

1 Introduction :

Le Machine Learning, également appelé apprentissage automatique, occupe une place centrale dans les domaines de la science des données et de l'intelligence artificielle. [18] [19] qui permet à un système informatique d'apprendre à partir de données et de réaliser des tâches sans être explicitement programmé. Le but de la machine Learning est de développer des modèles et des algorithmes qui permettent aux ordinateurs d'apprendre à partir des données, afin de les utiliser pour résoudre des problèmes complexes ou pour prendre des décisions. Le machine Learning est un domaine fascinant qui résulte de la combinaison de nombreuses disciplines telles que les statistiques, l'optimisation, l'algorithmique et le traitement du signal. Ce champ d'études est en constante évolution et a maintenant trouvé une place importante dans notre société. Bien qu'il soit utilisé depuis des décennies pour la reconnaissance automatique de caractères ou les filtres anti-spam, le machine Learning est maintenant utilisé pour de nombreuses autres applications, telles que la détection de la fraude bancaire, la recommandation de produits personnalisés, l'identification de visages dans les photos ou la traduction automatique de textes entre différentes langues. [20] [21] [22] dans ce chapitre en va voir définition et l'objectif de machine Learning et les algorithmes de ML , deep Learning , Les réseaux de neurones convolutifs , Réseau de neurones artificiels , Réseau neuronal récurrent.

2 Définition de Machine Learning:

Est une application de l'intelligence artificielle (IA) permet aux systèmes d'apprendre et de s'améliorer automatiquement en utilisant leur propre expérience, sans nécessiter une programmation explicite. [23] [24] Il s'agit d'un système ou d'une technique qui se sert principalement de données et d'algorithmes pour reproduire la façon dont les êtres humains apprennent, en améliorant continuellement sa précision. [25]

3 L'objectif de Machine Learning :

L'objectif principal du machine Learning est de permettre à des ordinateurs d'apprendre automatiquement à partir de données, sans être explicitement programmés pour chaque tâche. Le machine Learning utilise des algorithmes et des modèles mathématiques pour analyser les

données et en extraire des modèles et des tendances. En général, les objectifs du machine Learning sont les suivants :

- Classification : Prédire la classe ou l'étiquette d'un objet ou d'un événement donné, tel que la reconnaissance de visages ou l'identification de spam.
- Régression : Prédire une valeur numérique continue à partir de données, telles que la prédiction du prix des maisons ou des actions.
- Clustering : Regrouper les données en clusters ou en groupes similaires sans étiquettes préalables.
- Réduction de la dimensionnalité : Trouver une représentation de données avec moins de variables tout en conservant l'essentiel de l'information.
- Apprentissage par renforcement : Apprendre à prendre des décisions en interagissant avec un environnement, en recevant des récompenses pour des actions correctes et des sanctions pour des actions incorrectes.

Le machine Learning est utilisé dans de nombreuses applications, telles que la reconnaissance de la parole, la vision par ordinateur, la recommandation de produits, la prédiction de la fraude et la détection de spam.

4 Les algorithmes de Machine Learning :

Les algorithmes de Machine Learning sont des méthodes mathématiques et statistiques qui permettent aux ordinateurs d'apprendre à partir de données et d'exécuter des tâches sans qu'il soit nécessaire de les programmer explicitement à chaque étape . sont en mesure de prendre des décisions plus précises en se basant sur les données qui leur sont fournies. [26] Il existe plusieurs types d'algorithmes de Machine Learning, chacun étant conçu pour résoudre des problèmes spécifiques. Voici les types principaux :

4.1 l'apprentissage supervisé :

4.1.1 définition :

l'apprentissage supervisé est une technique de machine Learning qui utilise des données préalablement étiquetées ou annotées pour entraîner des modèles d'IA capables de prédire des résultats futurs . [27] Le processus d'apprentissage supervisé implique de guider la machine en lui montrant des exemples ou des données pour lui permettre d'accomplir une tâche spécifique.

4.1.2 Les applications de l'apprentissage supervisé :

Cette méthode est couramment utilisée dans de nombreuses applications telles que la reconnaissance vocale, la vision par ordinateur, les régressions et les classifications. Étant donné que la plupart des problèmes de Machine Learning et de Deep Learning font appel à l'apprentissage supervisé, il est crucial de bien comprendre son fonctionnement pour réussir dans ce domaine. [28] est employé dans divers domaines tels que la détection de courriers indésirables dans les emails [29], la gestion des chatbots et des voicebots [30], ainsi que dans la robotique . En outre, cette méthode favorise l'avancement des technologies embarquées spécifiquement conçues pour les véhicules autonomes [31] .

4.1.3 les algorithmes d'apprentissage supervisé :

4.1.3.1 Régression linéaire :

La régression linéaire est à la fois l'un des algorithmes d'apprentissage supervisé les plus populaires et l'un des plus simples à comprendre en statistique et en apprentissage automatique C'est une forme d'analyse prédictive élémentaire.

utilisés pour modéliser la relation entre une variable dépendante et une ou plusieurs variables indépendantes .Le but de l'algorithme est de trouver la ligne droite qui permet de mieux expliquer la variation de la variable dépendante en fonction des caractéristiques.

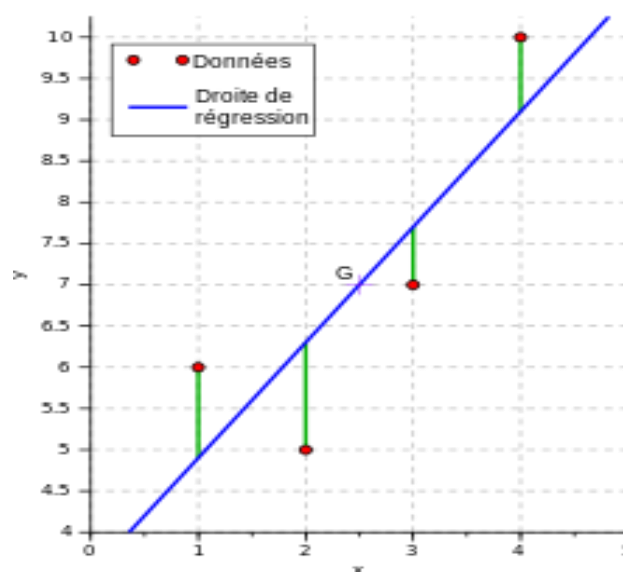


Figure 1 : La régression linéaire [43]

4.1.3.2 la régression logistique :

La régression logistique est un modèle mathématique qui permet d'estimer la probabilité d'occurrence d'un événement (variable dépendante) en fonction d'un ensemble de variables indépendantes ou explicatives. Ces variables représentent les facteurs qui peuvent potentiellement influencer l'événement en question. [32]

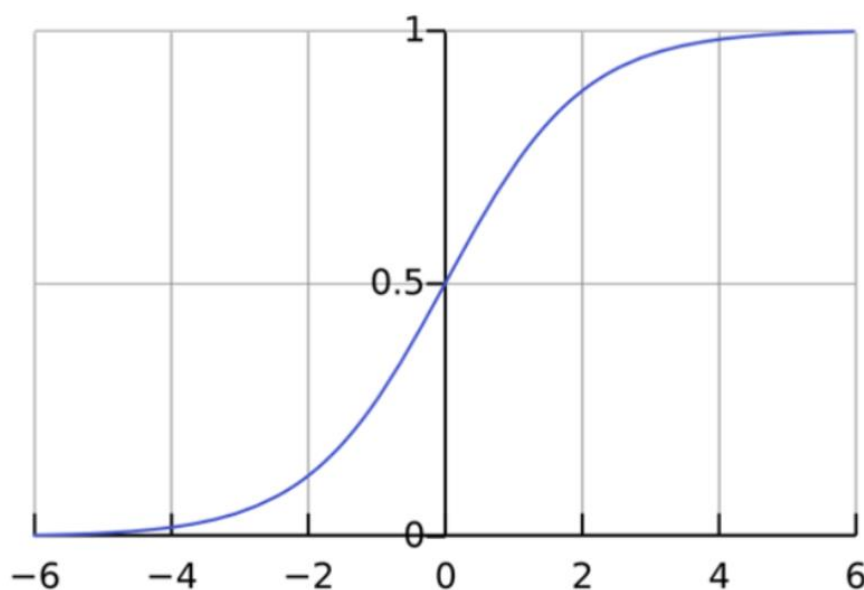


Figure 2: La régression logistique [33]

La régression logistique produit un résultat qui est une probabilité comprise entre 0 et 1. Lorsque cette probabilité dépasse 0,5, cela indique que l'événement est susceptible de se produire, tandis qu'une probabilité inférieure à 0,5 suggère que l'événement est peu probable. En utilisant ces probabilités, la régression logistique peut être utilisée pour construire des modèles de prédiction et de classification en optimisant les coefficients de régression.

La formule de la régression logistique est la suivante :

$$p = 1 / (1 + \exp(-z))$$

- p est la probabilité d'appartenance à la classe positive.
- exp est la fonction exponentielle.
- z est la somme pondérée des variables prédictives :

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$

- $\beta_0, \beta_1, \beta_2, \dots, \beta_k$ sont les coefficients de régression qui mesurent l'effet des variables prédictives sur la probabilité de la variable dépendante .

- X_1, X_2, \dots, X_k sont les valeurs des variables prédictives pour l'observation considérée.

4.1.3.3 SVM :

Les SVM (Support Vector Machine ou Machine à vecteurs de support) sont une catégorie d'algorithmes d'apprentissage automatique qui peuvent être utilisés pour résoudre des problèmes de classification, de régression et de détection d'anomalies. [34] .Dans les années 1990, les informaticiens russes Vladimir Vapnik et Alexey Chervonenkis ont conçu les premières machines à vecteurs de support en utilisant leur théorie statistique de l'apprentissage automatique appelée la théorie de Vapnik-Chervonenkis.

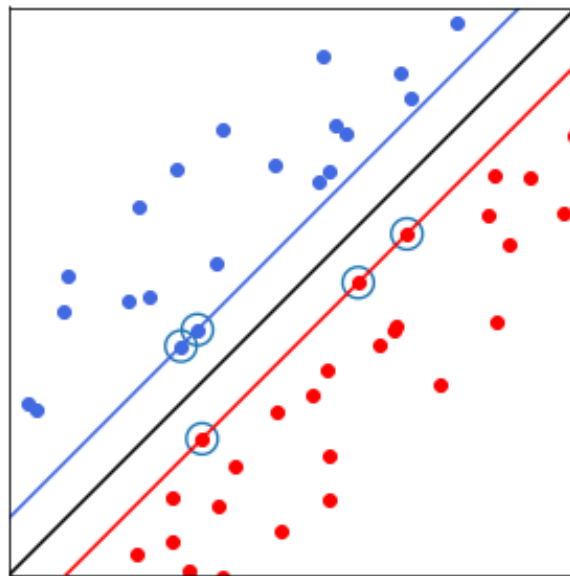


Figure 3: Machine à vecteurs de support [35]

Dans cette zone bidimensionnelle, la ligne noire représente la « frontière », tandis que les points encerclés, les plus proches de la frontière, sont appelés « vecteurs de support ». La distance entre la frontière et les lignes bleue et rouge est appelée la « marge ».

La notion de frontière implique que les données peuvent être séparées de manière linéaire, ce qui est rarement le cas. Pour résoudre ce problème, les SVM utilisent souvent des « noyaux », des fonctions mathématiques qui projettent les données dans un « feature space » (un espace vectoriel de plus grande dimension, comme illustré dans la figure ci-dessous). En outre, la

technique de maximisation de la marge renforce la robustesse du modèle face au bruit et assure ainsi une meilleure généralisation.

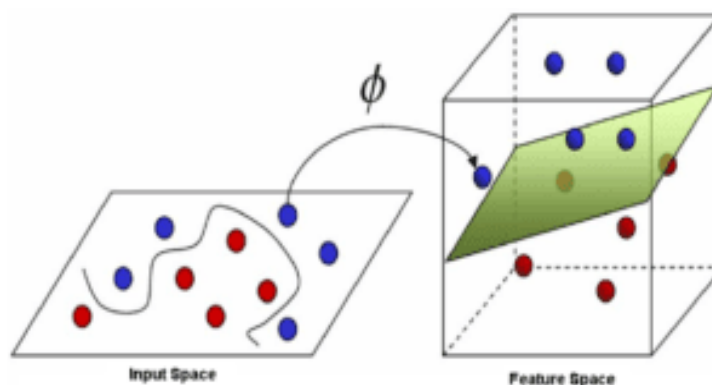


Figure 4: un espace vectoriel de plus grande dimension [42]

Les SVM utilisent une fonction noyau pour projeter les données dans un espace de dimension supérieure, ce qui facilite leur séparation linéaire.

Les SVM sont couramment utilisés dans divers domaines tels que la bioinformatique, la recherche d'informations, la vision par ordinateur et la finance, car contrairement aux réseaux de neurones, leur fonctionnement n'a pas besoin d'être compris pour les utiliser efficacement. En effet, il existe des jeux d'hyperparamètres par défaut pour la classification, la régression ou la détection d'anomalies, qui fonctionnent dans la plupart des cas.

L'un des principaux avantages des SVM est donc la simplicité de leur utilisation. De plus, le nombre d'hyperparamètres est très limité, se résumant au choix de la technique de régularisation (lasso ou régularisation RKHS (Reproducing Kernel Hilbert Space ou espace de Hilbert à noyau autoreproduisant), spécifique aux SVM) et du noyau (polynomiaux, Sobolev, RBF (Radial Basis Function, désigne souvent le noyau gaussien.), etc.). Parmi les algorithmes SVM, on peut citer le kernel ridge regression pour la régression et le one class SVM pour la détection d'anomalies.

D'après les données disponibles, il est généralement constaté que les performances des SVMs sont similaires, voire supérieures, à celles des réseaux de neurones ou des modèles de mélanges gaussiens. Cependant, il convient de noter que pour certaines tâches spécifiques, telles que la classification d'images, ces modèles peuvent être plus performants que les SVMs. En outre, l'utilisation d'un noyau RBF permet aux SVMs de devenir un "approximateur universel", ce qui signifie que, avec suffisamment de données, l'algorithme est capable de trouver la meilleure frontière possible pour séparer deux classes, à condition que cette frontière existe.

4.2 l'apprentissage non supervisé :

4.2.1 définition :

L'apprentissage non supervisé est une sous-discipline du domaine de l'apprentissage automatique, qui se concentre sur l'analyse et la classification de données non étiquetées sans la nécessité de la supervision d'un expert humain. [36] Les techniques courantes de l'apprentissage non supervisé comprennent la réduction de dimensionnalité, la classification non supervisée, le clustering, l'analyse en composantes principales (PCA) et les réseaux de neurones auto-encodeurs.

4.2.2 Les applications de l'apprentissage non supervisé :

L'apprentissage non supervisé a de nombreuses et diverses applications dans divers domaines. telles que le clustering, la réduction de la dimensionnalité, la détection des anomalies et l'association de règles. Ces techniques sont utiles pour l'analyse exploratoire des données, la segmentation de la clientèle, la détection de fraudes ou de comportements inhabituels, la recommandation de produits et la personnalisation de l'expérience utilisateur.

4.2.3 Les algorithmes de l'apprentissage non supervisé :

4.2.3.1 *K means* :

L'algorithme K-means, également connu sous le nom de méthode des K-moyennes, est un modèle d'apprentissage non supervisé qui permet de partitionner les données de manière non hiérarchique. [37]

Il s'agit d'un algorithme itératif qui fonctionne de la manière suivante :

- Sélectionnez le nombre de clusters (K) que vous souhaitez trouver.
- Sélectionnez aléatoirement K points de données pour être les centres de chaque cluster initial.
- Calculez la distance entre chaque point de données et les K centres de cluster.
- Affectez chaque point de données au cluster dont le centre est le plus proche.
- Recalculez les centres de chaque cluster en prenant la moyenne de tous les points de données qui lui sont assignés.
- Répétez les étapes 3 à 5 jusqu'à ce que les centres des clusters ne changent plus ou que le nombre maximum d'itérations soit atteint.

L'algorithme K-means cherche à minimiser la distance entre les points de données et leur centre de cluster respectif, tout en maximisant la distance entre les différents centres de cluster. Les résultats de l'algorithme dépendent donc fortement de la sélection initiale des centres de cluster, qui peut être effectuée de manière aléatoire ou en utilisant une méthode plus avancée telle que la méthode des k-means++.

K-means est couramment utilisé dans de nombreux domaines, tels que la segmentation de clients, la classification d'images et la détection d'anomalies. Cependant, il peut être sensible à la présence de points de données aberrants et ne fonctionne pas toujours bien avec des données de dimensions élevées.

4.2.3.2 Algorithme Apriori :

Les algorithmes Apriori sont des algorithmes utilisés en apprentissage automatique. exploite des ensembles d'éléments fréquents dans les bases de données transactionnelles pour dériver des règles d'association, qui peuvent être utilisées pour quantifier la force de la connexion entre deux objets. En examinant la fréquence à laquelle les articles se produisent simultanément dans les transactions, l'algorithme peut identifier les articles susceptibles d'être achetés ensemble, ce qui donne un aperçu des modèles de comportement des consommateurs. [38] L'objectif principal de cet algorithme est de découvrir des règles d'association fréquentes dans un ensemble de transactions.

L'algorithme Apriori fonctionne en deux étapes principales :

- Définir un seuil de support minimum : le seuil de support est le nombre minimum de fois qu'un motif doit apparaître dans l'ensemble de données pour être considéré comme fréquent.
- Identifier tous les motifs à un élément (itemset) dans l'ensemble de données : cela implique d'identifier tous les éléments individuels dans l'ensemble de données.
- Calculer la fréquence de chaque motif à un élément (itemset) : cela implique de parcourir l'ensemble de données et de compter le nombre de fois que chaque élément apparaît.
- Éliminer les motifs à un élément qui ne répondent pas au seuil de support minimum.
- Identifier tous les motifs à deux éléments possibles en utilisant les motifs fréquents à un élément restants.
- Calculer la fréquence de chaque motif à deux éléments possibles.

- Éliminer les motifs à deux éléments qui ne répondent pas au seuil de support minimum.
- Répéter les étapes 5 à 7 jusqu'à ce qu'il n'y ait plus de motifs fréquents.
- Générer des règles d'association à partir des motifs fréquents restants : cela implique de trouver tous les sous-ensembles d'un motif fréquent et de calculer la confiance de chaque règle d'association. Une règle d'association est considérée comme forte si sa confiance dépasse un seuil de confiance minimum.
- Retourner toutes les règles d'association fortes identifiées dans l'ensemble de données.

4.2.3.3 *clustering hiérarchique :*

Le clustering hiérarchique est une technique d'apprentissage non supervisé en machine Learning utilisée pour regrouper des données similaires en groupes ou en clusters. Cette méthode commence par considérer chaque point de données comme un cluster individuel et fusionne ensuite les clusters les plus similaires les uns aux autres jusqu'à ce qu'un seul cluster contenant toutes les données soit obtenu.

Le clustering hiérarchique consiste à diviser un ensemble de données de manière hiérarchique en groupes en combinant progressivement les deux groupes les plus proches à chaque étape. [39] . Le processus de clustering hiérarchique implique la construction d'un arbre de clusters, appelé dendrogramme, qui représente la hiérarchie des groupes. Il existe deux types de clustering hiérarchique : le clustering hiérarchique agglomératif et le clustering hiérarchique divisif.

4.3 **l'apprentissage semi supervisé :**

4.3.1 *définition :*

L'apprentissage semi-supervisé représente une solution équilibrée entre l'apprentissage supervisé et non supervisé en permettant de traiter de grandes quantités de données sans nécessiter l'étiquetage complet de celles-ci. Il bénéficie des avantages des deux types d'apprentissage mentionnés.

4.3.2 *les applications d'apprentissage semi supervisé :*

les applications de l'apprentissage semi-supervisé incluent la classification, la détection d'anomalies, la prédiction de liens, l'apprentissage actif et le clustering. Cette approche combine des données étiquetées et des données non étiquetées pour entraîner un modèle prédictif et améliorer la précision et la performance du modèle

4.3.3 Les algorithmes de l'apprentissage semi supervisé :

Il existe plusieurs algorithmes pour l'apprentissage semi-supervisé, notamment :

4.3.3.1 Label Propagation Algorithm :

L'algorithme de détection de communautés appelé LPA permet de détecter des groupes ou des clusters au sein de réseaux complexes représentés par des graphes. Comparé à d'autres algorithmes, il est à la fois plus rapide et plus efficace, et nécessite également moins d'informations initiales pour fonctionner. [40] [41]

Lors du premier passage de l'algorithme, une étiquette est attribuée à chaque nœud du graphe, puis ces étiquettes sont propagées lors des passages suivants pour former des groupes de nœuds avec des étiquettes similaires. Ces groupes d'étiquettes sont considérés comme des communautés et constituent le résultat de l'algorithme lorsque celui-ci atteint son point d'arrêt. [42].

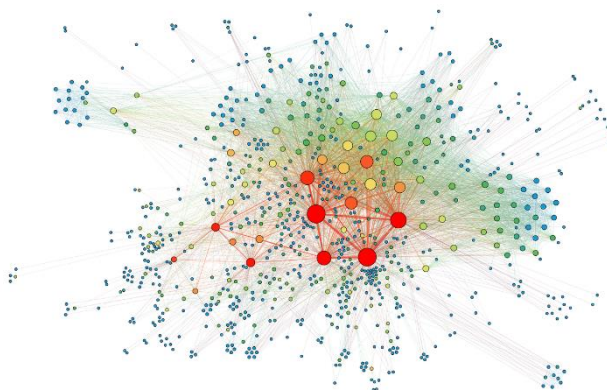


Figure 5: LPA [43]

4.3.3.2 L'apprentissage par co-enseignement (co-training) :

deux ou plusieurs classifieurs sont entraînés sur des ensembles de caractéristiques différents pour exploiter les informations mutuelles.

4.4 L'apprentissage par renforcement :

L'apprentissage par renforcement (reinforcement learning RL) est une branche de l'intelligence artificielle qui concerne l'apprentissage automatique basé sur la prise de décision et la récompense. Dans ce type d'apprentissage, un agent apprend à accomplir une tâche en interagissant avec un environnement et en recevant des récompenses ou des pénalités en

fonction de ses actions. Dans le cadre de l'apprentissage par renforcement, l'agent se sert d'une politique π pour sélectionner une action a . Le but de cette méthode est d'optimiser la politique d'action π de l'agent en utilisant des récompenses positives et négatives dans un contexte de jeu.

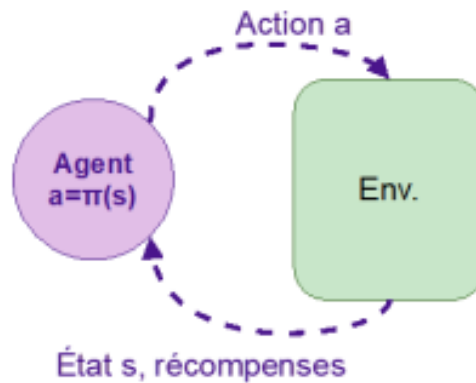


Figure 6: Schéma d'interaction entre l'agent et son environnement [44]

Pour pratiquer l'apprentissage par renforcement, il est essentiel de comprendre plusieurs concepts et mesures fondamentales, tels que :

Agent : un système ou un robot qui interagit avec son environnement et prend des actions en fonction de son état et de sa politique .

Action a : l'une des options disponibles pour l'agent lorsqu'il prend des décisions en fonction de son état .

État s : une situation spécifique dans laquelle l'agent se trouve à un moment donné .

Politique π : une stratégie qui définit le comportement de l'agent en fonction de son état actuel.

4.5 Deep learning :

Deep learning est un sous-ensemble du machine learning , sont composés de couches de réseaux neuronaux qui sont conçues de manière à imiter de manière approximative le fonctionnement du cerveau humain. [44]. En utilisant de vastes ensembles de données, l'entraînement des réseaux de neurones permet de configurer leurs paramètres. Cette configuration conduit à la création d'un modèle de deep learning qui, une fois entraîné, est capable d'analyser efficacement de nouvelles données, Les modèles de deep learning collectent des informations provenant de multiples sources de données et effectuent une analyse en temps réel, sans nécessiter d'intervention humaine. L'un des avantages clés de l'apprentissage profond est sa capacité à extraire des caractéristiques à partir de données brutes, sans nécessiter une connaissance experte approfondie du domaine. Cela signifie qu'il peut être utilisé pour résoudre

des problèmes dans des domaines variés tels que la médecine, la finance, les sciences de l'environnement et les sciences sociales.

4.6 Réseau de neurones artificiels (ANN):

Un réseau de neurones artificiels est un modèle mathématique inspiré du fonctionnement du cerveau humain. Il est constitué de nombreux neurones artificiels interconnectés, organisés en couches. Chaque couche de neurones transforme l'information d'entrée en une représentation plus abstraite, jusqu'à ce que l'information soit finalement classifiée ou régressée par la dernière couche. Le réseau de neurones artificiels utilise plusieurs processeurs qui fonctionnent simultanément et sont organisés en niveaux, Le premier niveau est responsable de la réception des données brutes en entrée, Les sorties d'informations transmises par le niveau précédent sont ensuite reçues par chaque niveau suivant, Le dernier niveau est chargé de produire les résultats du système, Pour traiter des problèmes complexes, il est nécessaire d'ajouter plus de niveaux au réseau. [45] Chaque neurone dans un réseau de neurones artificiels a une valeur qui détermine l'information qu'il peut transmettre au système. La fonction d'activation est utilisée pour calculer la sortie de chaque neurone, ce qui détermine le nombre de neurones qui doivent être activés pour résoudre un problème donné. Un algorithme est ensuite créé pour associer une sortie à chaque entrée.

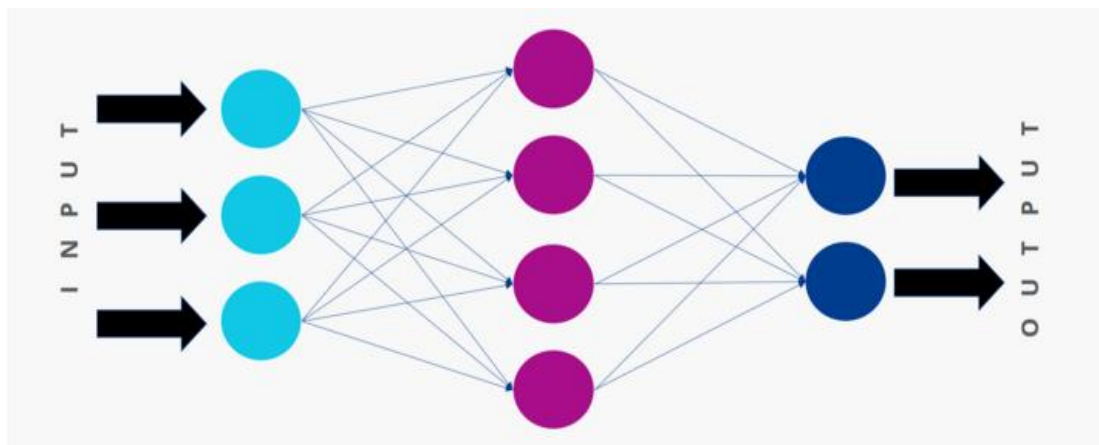


Figure 7 : un réseau de neurones artificiels [46]

4.6.1 Les réseaux de neurones convolutifs (CNN):

Les réseaux de neurones convolutifs sont directement inspirés du cortex visuel des vertébrés. On distingue deux parties, une première partie que l'on appelle la partie convolutive du modèle

et la seconde partie, que l'on va appeler la partie classification du modèle qui correspond à un modèle MLP (Multi Layers Perceptron).

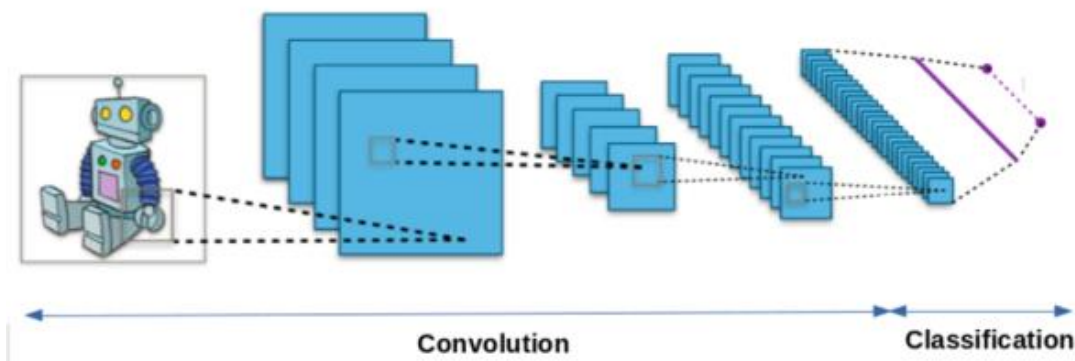


Figure 8: réseaux de neurones convolutifs [47]

C'est un réseau de neurones multicouches et plus précisément c'est un réseau profond composé de multiples couches qui en générale sont organisées en blocs (CONV → RELU → POOL). [48] Il existe quatre types de couches pour un réseau de neurones convolutif: la couche de convolution, la couche de pooling, la couche de correction ReLU et la couche fully-connected. [48]

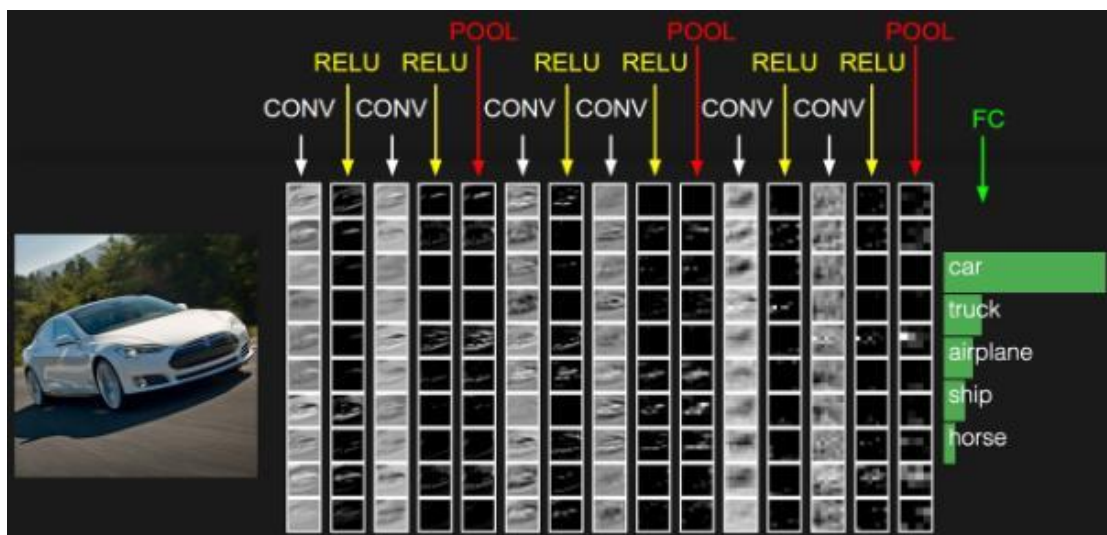


Figure 9: les couches de réseau de neurones [49]

4.6.2 Réseau neuronal récurrent (RNN):

Un réseau de neurones récurrent (RNN) est un type de réseau de neurones artificiels qui est capable de traiter des données séquentielles. Sa structure comporte au moins une connexion

récurrente qui forme une boucle ou un cycle. les signaux peuvent donc revenir en arrière (via une boucle ou un cycle) et alimenter des neurones d'une couche précédente ou de la même couche. [48] Les réseaux neuronaux récurrents présentent la capacité de traiter des données de tailles variables, les rendant particulièrement adaptés pour l'analyse de séquences temporelles ou chronologiques.[50] Il existe différents types de RNN, tels que les réseaux de neurones récurrents à mémoire courte (LSTM) et les réseaux de neurones récurrents à mémoire à court terme (GRU), et ont une grande variété d'applications dans de nombreux domaines, tels que le traitement du langage naturel, la reconnaissance d'images, la prédiction de séries temporelles et bien d'autres encore.

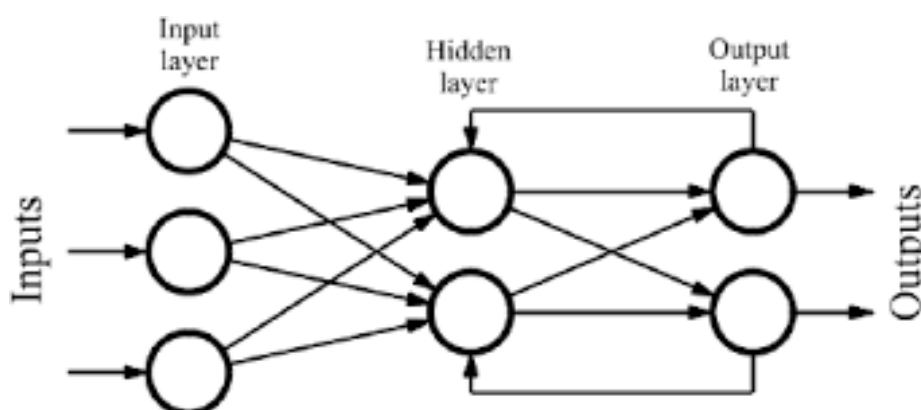


Figure 10 : Réseau de neurones récurrents [51]

5 Conclusion :

Dans ce chapitre nous avons vu Machine Learning et expliqué le fonctionnement des algorithmes d'apprentissage automatique les plus populaires, l'apprentissage automatique est un outil puissant pour faire des prédictions à partir de données. Cependant, il est important de se rappeler que l'apprentissage automatique est aussi bon que les données utilisées pour former les algorithmes. Afin de faire des prédictions précises, il est important d'utiliser des données de haute qualité qui sont représentatives des données du monde réel sur lesquelles l'algorithme sera utilisé. Dans le chapitre suivant nous allons présenter un état de l'art sur les travaux existants qui utilisent ML dans la négociation de SMA.

Chapitre 3 : Les travaux de ML dans la négociation de SMA

1 Introduction :

Dans le chapitre précédent, nous avons vu le ML et expliqué le fonctionnement des algorithmes d'apprentissage automatique les plus populaires, Et dans ce chapitre nous verrons les travaux existants qui utilisent ML dans la négociation de SMA. Les travaux de Machine Learning dans la négociation de SMA consistent généralement à entraîner des modèles de réseaux de neurones à partir de données de négociation passées, afin de prédire les comportements et les décisions des autres agents. Le ML peut également être utilisé pour améliorer la compréhension des relations entre les différents agents dans un SMA, en analysant les modèles de comportement et les interactions passées. Cette compréhension peut aider à élaborer des stratégies de négociation plus efficaces pour l'ensemble des agents et à maximiser les résultats de la négociation. Dans ce chapitre nous avons proposées une revue de l'état de l'art sur les travaux de ML dans la négociation Y compris Apprentissage par renforcement multi-agents pour la négociation avec des adversaires inconnus, Apprendre à négocier dans des systèmes multi-agents à l'aide de l'apprentissage par renforcement profond ,Apprentissage par renforcement profond pour la négociation (Learning from Experience and Dialogue Feedback) , Conception de mécanismes d'enchères avec apprentissage par renforcement dans des environnements multi-agents , l'apprentissage bayésien, régression non _ linéaire, estimation de la densité du noyau ,Enquête sur l'apprentissage dans les environnements multi-agents (Gérer la non-stationnarité).

2 Les travaux de Machine Learning:

Le Machine Learning est un domaine très utile dans la négociation de systèmes multi-agents car il permet aux agents de s'adapter et d'apprendre de manière autonome, ce qui peut améliorer considérablement les résultats de la négociation. Il y a plusieurs travaux existants qui utilisent la machine Learning dans la négociation système multi-agent (SMA). Voici quelques exemples:

2.1 Apprentissage par renforcement multi-agents pour la négociation avec des adversaires inconnus :

L'apprentissage par renforcement multi-agents (MARL) peut être utilisé pour développer des agents de négociation intelligents qui peuvent interagir avec des adversaires inconnus. La première étape consiste à définir les règles et les objectifs du jeu. Ensuite, les agents peuvent

apprendre indépendamment ou en coopération à l'aide d'algorithmes d'apprentissage par renforcement. Pour s'adapter aux différents adversaires, les agents peuvent utiliser des méthodes de Learning ou de gradient politique. Les performances du système peuvent être évaluées par des simulations avec différents adversaires. Les systèmes de négociation basés sur MARL ont des applications potentielles dans le commerce électronique, l'allocation des ressources et la résolution des conflits

2.2 Apprendre à négocier dans des systèmes multi-agents à l'aide de l'apprentissage par renforcement profond :

L'apprentissage par renforcement profond Learning pour apprendre à négocier dans un environnement de marché. Les agents apprennent à maximiser leur profit en interagissant avec d'autres agents.

L'apprentissage évolutif des politiques coordonnées des agents et de la communication entre les agents dans les SMA est un problème ouvert de longue date. Nous nous attaquons à ce problème en développant des algorithmes pour la RL profonde multi-agents, dans lesquels plusieurs agents apprennent à communiquer et à interagir de manière optimale pour atteindre un objectif spécifique, Alors que RL profond a permis l'évolutivité à de grands espaces d'état, l'objectif de MARL doit permettre une évolutivité efficace du nombre d'agents là où l'espace de décision conjointe le ferait autrement. Être insoluble pour des approches centralisées. [52]

2.3 Apprentissage par renforcement profond pour la négociation (Learning from Experience and Dialogue Feedback):

L'apprentissage par renforcement profond (DRL) est un sous-domaine de l'apprentissage automatique qui peut être appliqué à des scénarios de négociation. L'approche consiste à former un agent à apprendre de l'expérience et du retour d'expérience du dialogue (Learning from Experience and Dialogue Feedback) afin de prendre des décisions optimales. Dialogue feedback peuvent être utilisés pour améliorer les performances de l'agent. L'agent interagit avec l'environnement de négociation et apprend des résultats de ces interactions pour maximiser sa récompense. Les décisions de l'agent sont prises à l'aide d'un réseau de neurones et d'une politique qui équilibre l'exploration et l'exploitation .DRL peut gérer des scénarios de négociation complexes avec de nombreuses variables. En fin de compte, le DRL pour la négociation à le potentiel de révolutionner la façon dont nous abordons les négociations dans divers contextes.

2.4 Conception de mécanismes d'enchères avec apprentissage par renforcement dans des environnements multi-agents:

La conception du mécanisme d'enchères implique la conception des règles et procédures d'une enchère pour atteindre certains objectifs, tels que la maximisation des revenus du commissaire-priseur, la garantie d'une allocation efficace des biens et la promotion d'une concurrence loyale entre les soumissionnaires. Dans les environnements multi-agents, où plusieurs agents participent à l'enchère, la conception de mécanismes d'enchères efficaces peut être difficile en raison de la complexité des interactions entre les agents.

Dans cette conception de jeu, les agents s'affrontent pour obtenir un produit P dans une enchère. Chaque agent a une évaluation privée pour le produit vendu. L'action définie pour les agents consiste de « offrir un nouveau prix » (le prix est une option disponible dans l'enchère qui peut être croissante ou décroissante pour les différents types d'enchères) et "arrêter d'enchérir". Lorsqu'une enchère se termine, la récompense pour le gagnant est la différence entre le prix du gagnant et l'appréciation de l'agent ; le perdant n'obtient aucun résultat. La figure ci-dessous illustre un cas simpliste pour une enchère en apprentissage par renforcement. Les agents ont l'espace d'action de "Bid" et "Stop". Enchérir signifie l'action initiale pour intervenir dans l'enchère, ainsi que l'augmentation du prix actuel du produit. L'action "Stop", d'autre part, a le sens de reposer l'enchère. En fonction du type d'enchère, l'état suivant est déterminé par le comportement des agents. La récompense finale est la différence entre l'évaluation privée du produit et le prix de clôture de l'enchère. La récompense sera positive lorsque le prix est fixé en dessous du niveau d'expertise ; sinon, ce sera 0 ou négatif. Dans le scénario de cas où cette étude suggère, il n'y a pas de frais d'entrée ou d'enchères. Par conséquent, si un agent dessine l'enchère à n'importe quelle étape, la récompense sera nulle. [53]

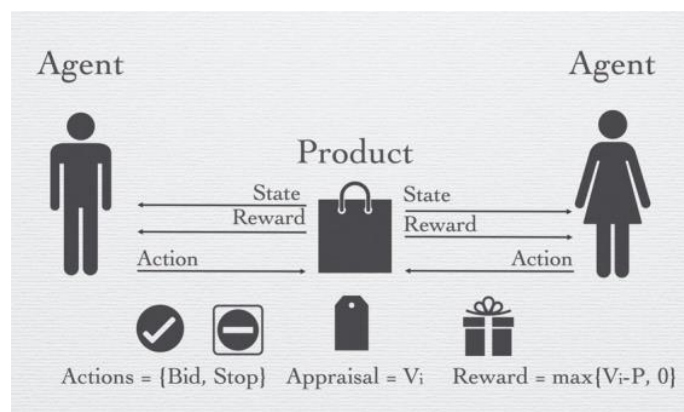


Figure 11: Un mécanisme d'apprentissage par renforcement pour une enchère conception [54]

2.5 Apprentissage bayésien :

Une méthode d'apprentissage bayésienne est présentée pour évaluer la valeur de la réservation en utilisant les données de négociations passées. Cette méthode se concentre sur une seule issue quantitative qui suppose que les agents ont des préférences opposées. Avant la négociation, un ensemble d'hypothèses $H = \{h_1 \dots \dots h_n\}$ sur la valeur de réservation de l'adversaire est généré. Chaque hypothèse H_i est de la forme $r_v = v_i$, où v_i est l'une des valeurs possibles pour la valeur de réservation de l'adversaire r_v . Les hypothèses, les valeurs v_i et leur vraisemblance a priori sont déterminées à partir des connaissances du domaine tirées des négociations précédentes. Pendant la négociation, l'apprentissage bayésien est utilisé pour mettre à jour les probabilités des hypothèses en fonction du comportement observé et des connaissances du domaine disponibles. L'idée sous-jacente est qu'une offre initiale est susceptible d'être loin de la valeur de la réservation. La valeur de la réservation est estimée en utilisant une somme pondérée des hypothèses en fonction de leur probabilité. [55]

2.6 Estimation de la densité du noyau :

La probabilité d'acceptation calculée par interpolation ne peut être directement appliquée à un paramètre de négociation à problèmes multiples, car l'espace de préférence à problèmes multiples est structuré différemment de l'espace de préférence à problème unique avec des préférences opposées. Pour résoudre ce problème, les auteurs de cette étude ont utilisé une base de données de négociations impliquant des négociateurs humains avec des profils de préférences connus. Pendant la négociation, la position de profil de préférence était supposée connue, et les auteurs ont utilisé l'estimation de la densité du noyau pour estimer la probabilité d'acceptation de toutes les autres offres. Les auteurs ont montré que les préférences de négociation d'un adversaire dans les négociations bilatérales multi-issues peuvent être efficacement apprises en utilisant l'estimation de la densité du noyau. Plus précisément, ils ont appliqué cette méthode au modèle de compromis de Faratin et ont démontré que cela peut améliorer l'efficacité des négociations en termes d'utilité et de délai d'accord. Contrairement à d'autres méthodes d'apprentissage, l'estimation de la densité du noyau ne nécessite pas d'hypothèses explicites sur la relation entre le temps, l'historique des négociations et les préférences de l'adversaire.

2.7 Régression non _ linéaire :

Hou propose une méthode innovante pour évaluer la tactique de l'adversaire dans une négociation portant sur un seul problème quantitatif, impliquant des délais privés. Selon cette

méthode, il est supposé que l'adversaire utilise une tactique qui dépend de facteurs tels que le temps, le comportement ou les ressources. La méthode de régression non linéaire est employée pour estimer le type de stratégie utilisé, parmi les trois existantes, ainsi que les paramètres associés à la tactique, y compris la valeur de réservation. Cette approche est similaire à celle d'Agrawal et Chari, qui utilisent une fonction exponentielle pour modéliser la fonction de décision de l'adversaire. Si le délai est connu du public, la méthode de Haberland peut être utilisée pour estimer la valeur de réservation de l'adversaire, en supposant qu'il utilise une tactique dépendante du temps. Yu et al. ont combiné la régression non linéaire avec l'apprentissage bayésien dans leur modèle pour améliorer la fiabilité des estimations de la valeur de réserve de l'adversaire et du délai. Ils ont postulé que l'adversaire mettait en œuvre une tactique temporelle avec des paramètres inconnus, et qu'à chaque tour, ces paramètres étaient évalués à travers une régression non linéaire. Par la suite, ils ont employé cette évaluation pour élaborer un ensemble plus fiable d'hypothèses sur la valeur de la réserve et le temps d'attente de l'adversaire en utilisant une méthode d'apprentissage bayésien.

2.8 Enquête sur l'apprentissage dans les environnements multi-agents (Gérer la non-stationnarité):

L'apprentissage multi-agents est confronté au défi majeur d'apprendre une réponse optimale face au comportement des autres agents, qui peuvent être des entités non-stationnaires. Cette situation rend difficile l'atteinte de la cible pour l'apprentissage, car les autres agents peuvent ajuster leur stratégie de manière constante. Cette enquête présente une vue d'ensemble cohérente des travaux qui utilisent des outils de la théorie des jeux, de l'apprentissage par renforcement et des bandits multi-armés pour traiter cette non-stationnarité provoquée par les adversaires. Elle examine les principales approches des algorithmes pour modéliser et faire face à cette non-stationnarité, et propose un nouveau cadre pour y répondre. L'apprentissage multi-agents a reçu beaucoup d'attention ces dernières années et certaines enquêtes précédentes ont émergé avec des motivations et des résultats différents. L'apprentissage multi-agents a reçu beaucoup d'attention ces dernières années et certaines enquêtes précédentes ont émergé avec des motivations et des résultats différents.

- Shoham et al.(2007) présenté une étude générale de l'apprentissage multi-agents fournissant quelques questions fondamentales intéressantes et identifier cinq agendas différents dans cette communauté de recherche. [56]

- Tuyls et Weiss (2012) présenté une vue d'ensemble sur le problème de l'intelligence artificielle de l'apprentissage multi-agents, en identifiant les étapes franchies par la communauté et mentionnant les défis ouverts à l'époque. [57]
- Anait et Luke (2005) ont présenté une analyse approfondie des algorithmes d'apprentissage coopératif multi-agents, en les divisant en deux catégories : apprenant unique dans un problème multi-agents (apprentissage en équipe) et plusieurs apprenants (apprentissage simultané).[58]

L'apprentissage par renforcement est une méthode d'apprentissage automatique, contient plusieurs algorithmes ou techniques d'apprentissage parmi eux l'algorithme Q-Learning.

3 Algorithme Q-Learning :

3.1 Définition :

Est-ce qu'un algorithme d'apprentissage par renforcement populaire existe qui ne repose pas sur le modèle et ne fait pas appel à l'"équation de Bellman" ci-dessous :

$$v(s) = \left[R(s, a) + \gamma \sum_{s'} p(s, a, s') v(s') \right]$$

Équation 1: équation de Bellman

L'objectif principal de cet algorithme est d'acquérir des stratégies permettant aux agents de déterminer quand prendre des actions afin de maximiser les récompenses. Dans notre situation, nous cherchons spécifiquement à maximiser l'efficacité de notre agent négociateur.

Il apprend la fonction de valeur Q (s, a) ; ce qui signifie à quel point il est bon d'agir 'a' à un état 's', dans notre cas veut dire quel offre est le meilleur parmi les offres proposées par l'adversaire. Le modèle permet une compréhension rapide du processus d'apprentissage de nombreuses techniques et permet de déterminer la meilleure action à prendre dans un contexte spécifique.[59]

L'organigramme suivant illustre la fonction du Q-Learning :

:

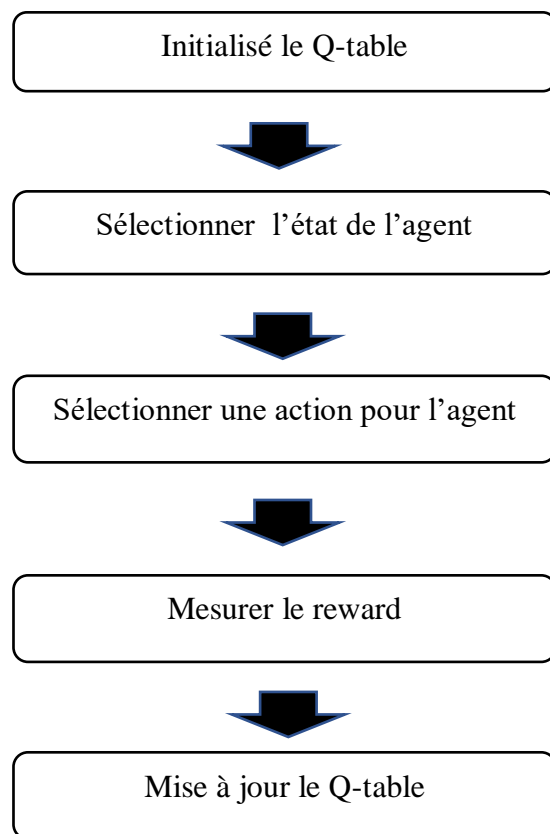


Figure 12: Explication de la fonction de Q-learning

3.2 Le Q-table :

Pendant le Q-learning, une table ou une matrice Q est construite. Cette table enregistre les paires d'état et d'action (s, a) et initialise leurs valeurs à zéro. Après chaque interaction, la table est mise à jour et les valeurs Q sont stockées. L'agent de renforcement utilise cette table Q comme référence pour sélectionner la meilleure action en fonction des valeurs Q.

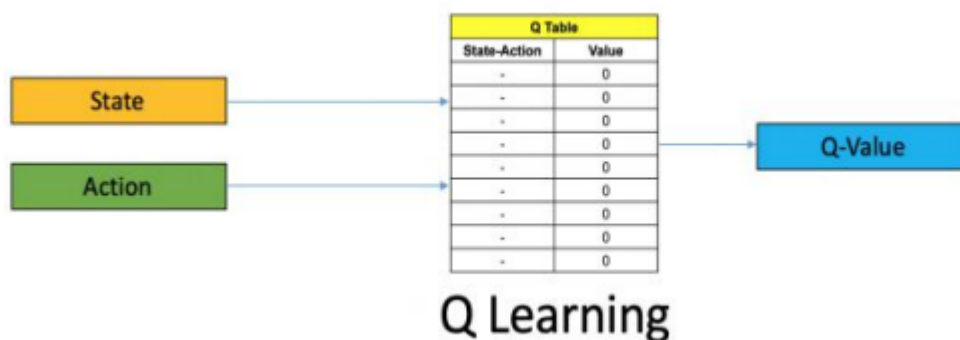


Figure 13: . Le Q-table [60]

3.3 Le principe :

Le principe de l'algorithme Q-Learning, défini formellement par la figure 12, est de mettre à jour itérativement, à la suite de chaque transition (s, a, r) la fonction de valeur courante Q pour le couple (s, a) ou s représente l'état courant, a l'action sélectionnée et réalisée, s' l'état résultant et r la récompense immédiate. Cette mise à jour se fait sur la base de l'observation des transitions instantanées et de leur récompense associée. [61]

```
Initialiser  $Q[s,a]$ 
Répéter un nombre  $N$  de fois
  initialiser l'état  $s$ 
  répéter
    choisir action  $a$  depuis  $s$  en utilisant  $Q$  et un peu d'aléatoire
    exécuter l'action  $a$ 
    observer la récompense  $r$  et l'état  $s'$ 
    mettre à jour  $Q$  :
     $Q[s, a] := Q[s, a] + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ 
     $s := s'$ 
  jusqu'à ce que  $s$  soit l'état terminal ou pour un nombre de boucles max
```

Figure 14: algorithme Q-Learning [62]

Dans cet algorithme, Le taux d'apprentissage $\alpha (s, a)$ est propre à chaque couple (s, a) , et décroît vers 0 à chaque passage. Et γ facteur de remise qui prendre la récompense suivante . Ces hyper paramètre α et γ leur valeur entre $[0,1]$.

Voici les étapes détaillées de l'algorithme de Q-Learning :

Initialisation : Initialisez la fonction de valeur Q pour tous les états et actions possibles avec des valeurs arbitraires ou à zéro.

Interaction : L'agent interagit avec l'environnement en suivant ces étapes :

- Observation de l'état : L'agent observe l'état actuel s de l'environnement.
- Sélection de l'action : L'agent choisit une action a à prendre dans l'état s en utilisant une politique d'exploration-exploitation. Pendant l'exploration, l'agent choisit des actions aléatoires pour explorer l'environnement. Pendant l'exploitation, l'agent choisit les actions avec les valeurs Q les plus élevées.

- c. Exécution de l'action : L'agent prend l'action a et l'environnement passe à l'état suivant s' et l'agent reçoit une récompense r pour avoir pris l'action a dans l'état s .
- d. Mise à jour de la fonction de valeur Q : La fonction de valeur Q est mise à jour en utilisant l'équation suivante :

$$Q(s, a) = r + \gamma \max(Q(s', a'))$$

Équation 2: équation de mise à jour de Q

- e. Déplacement vers l'état suivant : L'agent met à jour son état actuel $s \leftarrow s'$.
- f. Répétez les étapes a à e jusqu'à ce que la convergence soit atteinte.

Utilisation de la fonction de valeur Q : Utilisez la fonction de valeur Q pour déterminer la politique optimale. La politique optimale consiste à sélectionner l'action ayant la valeur Q la plus élevée pour chaque état.

4 Conclusion :

Dans ce chapitre nous avons vu les travaux existants qui utilisent ML dans la négociation de SMA tel que la conception de mécanismes d'enchères, l'apprentissage profond, l'adaptation à des adversaires inconnus, et l'algorithme de l'apprentissage par renforcement qui s'appelle le Q-Learning.

Le prochain chapitre présentera une approche de modélisation ou conception émergente pour la création d'un agent négociateur basé sur « Apprentissage par renforcement ».

Chapitre 4 : conception de notre agent

1 Introduction :

Dans le chapitre précédent, nous avons exposé diverses techniques de modélisation de Machine Learning dans le cadre de négociations multi-agents. Dans ce chapitre, nous allons construire un agent négociateur en utilisant la technique d'apprentissage par renforcement. Cette méthode se concentre sur la prédiction des actions de l'adversaire, ce qui est plus complexe que de déterminer ses intentions. Dans ce chapitre, nous commencerons par une brève présentation de notre agent négociateur, puis nous aborderons l'approche d'apprentissage par renforcement. Enfin, nous proposerons une conception de l'agent négociateur, ainsi que sa stratégie d'apprentissage, d'acceptation/refus, d'offre et de concession.

2 Description de notre agent :

Ce mémoire se concentre sur la modélisation d'un agent capable de négocier avec un adversaire et de maximiser son utilité pour remporter le tour de négociation. Nous utiliserons une stratégie d'apprentissage par renforcement pour permettre à l'agent de prédire les actions de son adversaire et de lui proposer des offres à la fois plus utiles pour lui-même et satisfaisantes pour l'adversaire.

La capacité d'un agent à réagir aux événements externes, son autonomie dans l'action sans intervention extérieure, sa rationalité dans l'utilisation de ses connaissances antérieures et de ses perspectives futures pour maximiser sa performance, la possession d'un ou plusieurs objets, la perception de son environnement et la possibilité d'agir en fonction de ses résultats, ainsi que son tempérament égoïste qui le pousse à privilégier son propre intérêt. [63]

3 Conception de notre agent :

La plateforme GENIUS sera utilisée pour mener des expériences visant à simuler des négociations automatisées entre des agents intelligents, dans le but de modéliser l'adversaire lors d'une négociation multi issues non linéaire, tel que décrit dans ce mémoire. Le produit cartésien des issues représente l'espace de résultats d'une négociation présenté sous forme :

$\Omega = \{W_1 \dots \dots \dots W_N\}$ ou W_i , L'ensemble des résultats envisageables est représenté par le terme "résultats possibles", et leur nombre total est indiqué par N.

Chaque agent associe une utilité à chaque résultat w en fonction de son propre profil de préférences. noté par $U_A \in [0, 1]$ de l'agent A , l'espace de résultat envisageables définit un scénario de négociation avec le profil de préférence. Chaque agent a une valeur minimale d'utilité, appelée "valeur réservée", en dessous de laquelle il ne sera jamais disposé à accepter un résultat. Cette valeur est notée β . Trouver un cadre d'apprentissage commun pour renforcer les négociations multi-issues s'avère être une tâche complexe. L'un des principaux défis consiste à trouver un moyen de représenter de manière concise un espace d'états et d'actions partagé, tout en permettant au concepteur d'agents de le personnaliser librement pour s'adapter à divers paramètres de négociation.

4 les stratégies de négociation :

La stratégie de négociation est un plan d'action qui vise à atteindre les objectifs fixés lors d'une négociation. Elle implique la préparation minutieuse de la négociation, la détermination des objectifs souhaités, ainsi que des stratégies et tactiques spécifiques pour les atteindre. La stratégie de négociation peut également inclure des plans d'action pour gérer les conflits et les objections, ainsi que des plans de communication pour établir une relation de confiance avec l'autre partie. En somme, la stratégie de négociation est un guide pour maximiser les résultats d'une négociation tout en préservant une relation positive entre les parties impliquées.

Notre stratégie de négociation qui comprend une approche d'apprentissage pour comprendre les besoins et les motivations de l'autre partie, une approche d'acceptation pour être ouvert aux idées de l'autre partie, une approche de proposition pour présenter nos propres idées, et une approche de concession pour trouver un terrain d'entente qui convienne à toutes les parties impliquées.

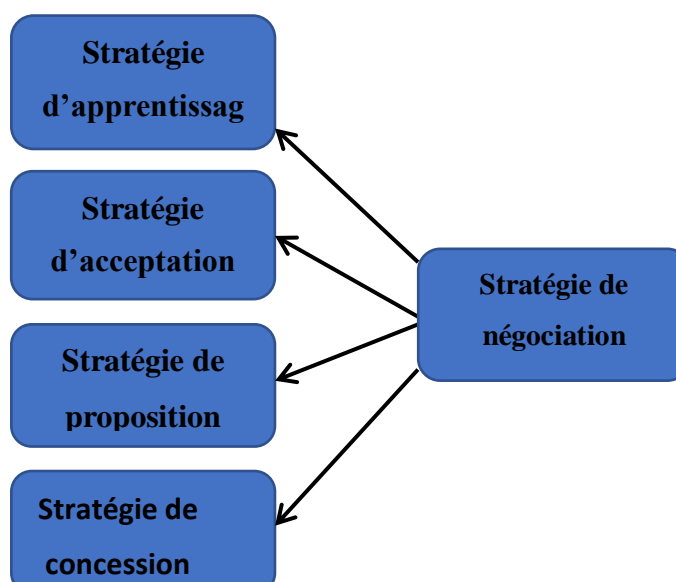


Figure 15: Stratégie de négociation

4.1 Stratégie d'apprentissage :

Notre approche d'apprentissage pour notre agent se divise en quatre étapes principales est la mémorisation, le renforcement, l'évaluation et la décision. Le modèle fonctionne selon cette structure organisée.

4.1.1 l'étape de mémorisation :

La mémoire permet de conserver la valeur d'utilité actuelle en utilisant une Q-table de dimension composée du nombre d'états possibles (S) et du nombre d'actions possibles (A). Ainsi, elle enregistre les valeurs d'utilité $Q(s, a)$ pour toutes les combinaisons possibles des situations S et des actions A.

4.1.2 L'étape de renforcement :

La fonction de renforcement du Q-Learning repose sur des algorithmes itératifs visant à optimiser la valeur de la fonction d'utilité pour chaque paire état-action (s, a) afin d'obtenir la meilleure stratégie de contrôle. Ainsi, son objectif est d'améliorer l'estimation de gain $Q(s, a)$ pour un état "s" et une action "a" en utilisant l'équation de mise à jour du Q-Learning.

4.1.3 L'étape d'évaluation :

Le module d'évaluation vise à relier la mémoire et les fonctions de prise de décision en fournissant la fonction de décision, qui représente l'estimation actuelle du retour attendu pour chaque action dans l'état actuel du système.

4.1.4 L'étape de décision :

Nous avons constaté que l'algorithme d'apprentissage Q converge vers la direction optimale en respectant la contrainte de parcourir tous les états et actions possibles de manière indéfinie (via la Q-table). L'agent doit constamment choisir quelle commande exécuter, et donc trouver un compromis entre exploitation et exploration. Pour ce faire, nous avons opté pour une fonction de décision ϵ -greedy, qui permet de privilégier les choix optimaux la plupart du temps tout en conservant un certain degré d'exploration.

4.2 Stratégie d'acceptation :

La stratégie d'acceptation est une technique de négociation qui implique d'accepter une offre ou une proposition faite par l'autre partie dans le cadre d'une négociation. Cette stratégie peut être

utilisée dans une variété de contextes de négociation, tels que les négociations commerciales, les négociations de contrat, les négociations salariales, etc. basée sur la combinaison de condition de temps ($t > T$) et utilité (U_A), elle est représentée comme suit :

```

Tant que  $t > T$  faire :
  Si  $U_A(W_{B \rightarrow A}^t) > d$  alors
    → A := accepte l'offre de l'opposant ;
  Sinon si [ $\alpha U_A(W_{B \rightarrow A}^t) > U_A(W_{A \rightarrow B}^{t+1})$ ]
  Si non A := refus l'offre de l'opposant ;

```

Figure 16: La stratégie d'acceptation [64]

4.3 Stratégie de proposition :

Notre agent utilisera la stratégie de proposition, qui implique de comprendre les préférences de l'adversaire et de proposer une offre en conséquence. Tout d'abord, nous allons fixer une valeur $d=0.9$ pour représenter l'utilité objective de notre offre. Ensuite, nous calculerons la distance euclidienne entre l'utilité de chaque offre et la dernière offre proposée par l'adversaire. Nous sélectionnerons la valeur "min-espace-euc" avec $d=0.9$ pour déterminer le moment où notre agent proposera une offre. Le pseudo-code suivant résume cette stratégie :

```

Tant que  $U_A(W_{B \rightarrow A}^t) < d$  et  $t > T$  faire :
  /*d la valeur d'utilité demandée */
  d := 0.9 ;
  → A := effectuer un choix pour proposer la plus proche offre de la
    dernière offre d'adversaire ;
  (Calculer la distance euclidienne)
  → A := choisir le résultat min de l'espace euclidienne ET  $d=0.9$  ;
  →  $U_A(W_{A \rightarrow B}^{t+1})$ .

```

Figure 17: la stratégie de proposition [64]

4.4 La stratégie de concession :

La stratégie de concession dans la négociation consiste à faire des compromis et des concessions afin de parvenir à un accord satisfaisant pour toutes les parties. Les agents qui utilisent cette stratégie offrent souvent des concessions mineures pour encourager l'autre partie à faire de même. Nous formulons une condition qui permet de faire des concessions en fonction du temps. Selon cette condition, lorsque la deadline ne représente plus que 10% du temps initial, l'agent commence à faire des concessions de la manière suivante : Il ajuste la valeur de l'utilité objective pour qu'elle corresponde à la valeur de l'utilité réservée en la diminuant si nécessaire.

$$\left\{ \begin{array}{l} \text{Tant que } d > \beta \text{ et } t \leq 10\%T \text{ faire} \\ d := d - 10\% d ; \end{array} \right.$$

Figure 18: stratégie de concession [64]

5 Conclusion :

Dans Ce chapitre expose les principes fondateurs de notre modèle de conception pour un agent négociateur. Le chapitre suivant vise à confirmer la validité de notre modèle de conception en le testant sur la plateforme de simulation de négociation automatique GENIUS.

Chapitre 5 : Implémentation et résultats

1 Introduction :

Dans le chapitre précédent, nous avons exposé les fondements de notre modèle pour concevoir un agent négociateur. Dans ce chapitre Nous avons dédié l'analyse comportementale de l'agent sur négociation multi issus non linéaire dans la plateforme GENIUS. Dans ce chapitre, nous expérimenterons l'un des algorithmes d'apprentissage par renforcement pour la négociation multi-issus non linéaire en effectuant des tests avec divers agents compatibles sur la plateforme utilisée. Nous présenterons également les logiciels qui ont été utilisés.

2 La plateforme GENIUS :

(Nous avons opté pour l'utilisation de la version 7.1.8 tout au long de cette mémoire.) GENIUS (Generic Environment for Negotiation with Intelligent multi-purpose Usage Simulation) .GENIUS est le premier environnement de ce genre qui soutient la conception de stratégies pour les négociateurs automatisés et facilite également le processus d'évaluation du mandataire. Nous estimons que cet environnement est extrêmement utile pour les concepteurs d'agents et peut jouer un rôle central dans le processus de conception des agents automatisés. Bien que les agents puissent être conçus avec n'importe quelle méthodologie d'ingénierie logicielle orientée agent, GENIUS propose une approche conviviale qui permet aux concepteurs de se concentrer sur le développement de stratégies pour la négociation dans un environnement ouvert doté de fonctions utilitaires multi-attributs. GENIUS se concentre sur la négociation bilatérale, c'est-à-dire la négociation entre deux parties ou agents A et B. Les agents négocient sur des questions appartenant à un domaine de négociation, où chaque question est associée à une gamme d'alternatives ou de valeurs. Le résultat de la négociation consiste en un mappage de chaque question à une valeur, et l'ensemble de tous les résultats possibles est appelé l'espace de résultat, noté Ω . L'espace de résultat est fixe et connu de toutes les parties pendant toute la durée d'une session de négociation.

Nous supposons également que les deux parties ont des préférences qui sont définies par un profil de préférences sur Ω . Ces préférences peuvent être modélisées à l'aide d'une fonction d'utilité normalisée U , qui associe un résultat possible $\omega \in \Omega$ à un nombre réel compris entre 0 et 1. [65]

2.1 Présentation des interfaces du GENIUS :

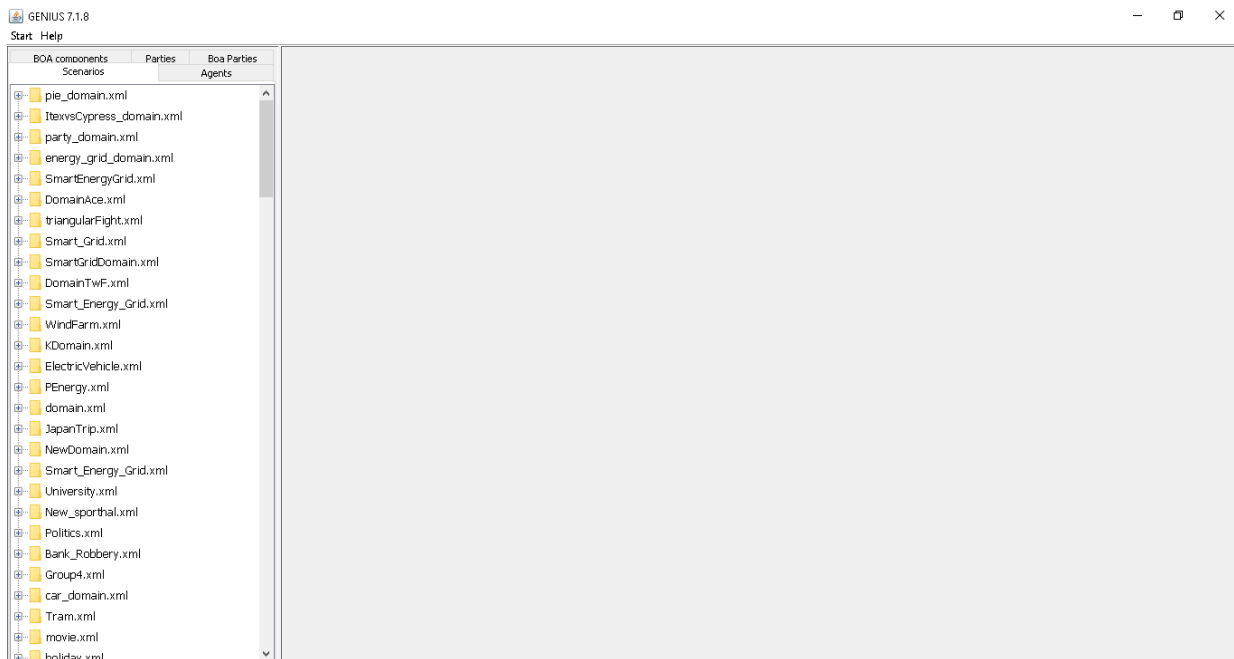


Figure 19: GENIUS juste après le démarrage, à la gauche on a le panneau de composants à la droite Le panneau d'état.

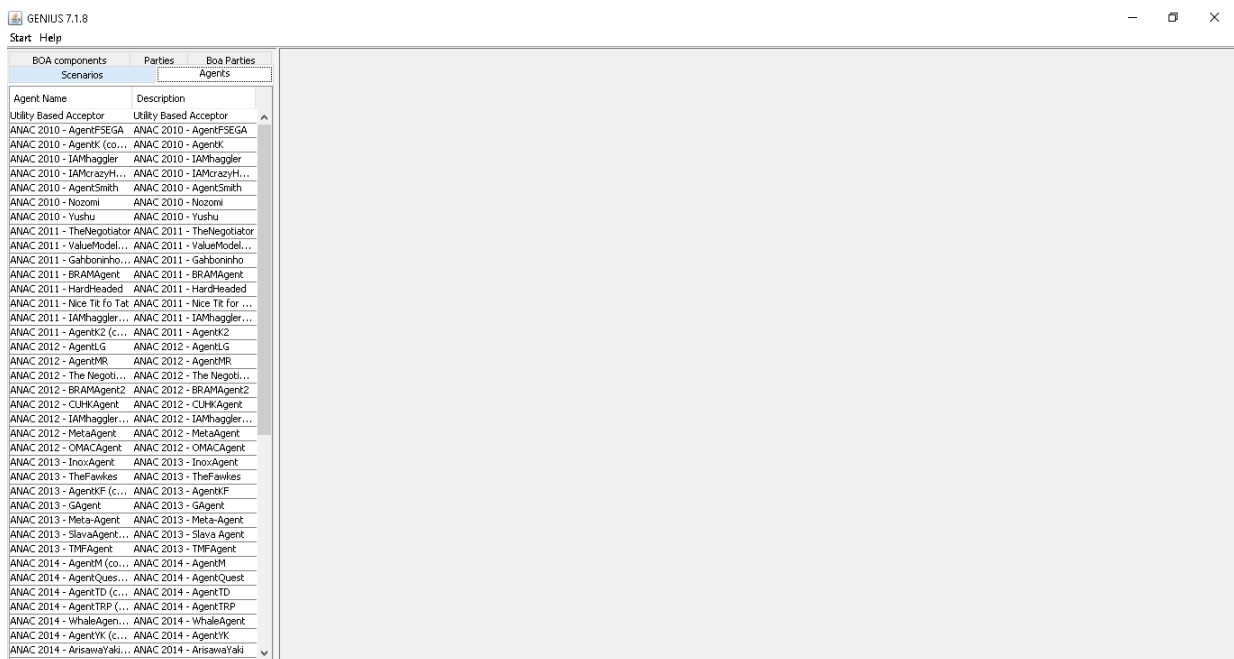


Figure 20: Les agents intégrés avec GENIUS.



Figure 21: : composants BOA ; offrant à l'utilisateur la possibilité de créer et d'appliquer des composants nouvellement développés utiliser une interface utilisateur graphique.

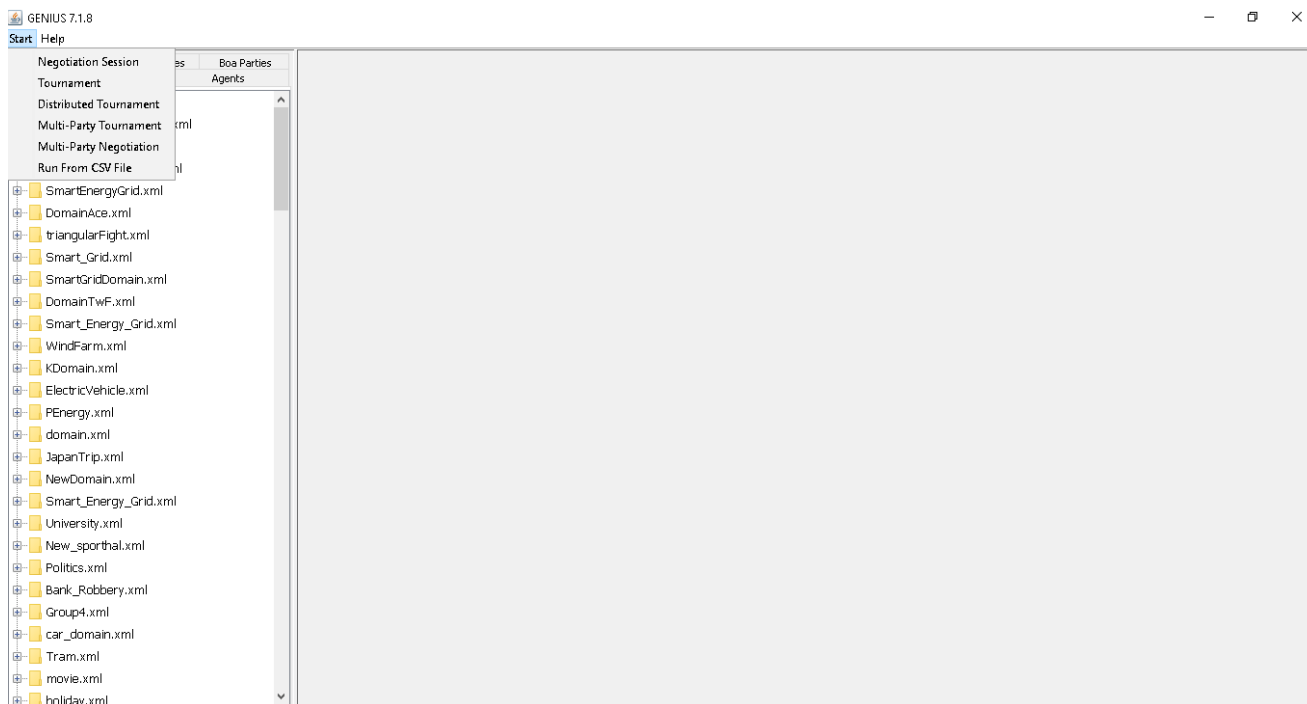


Figure 22: Création d'une négociation session simple.

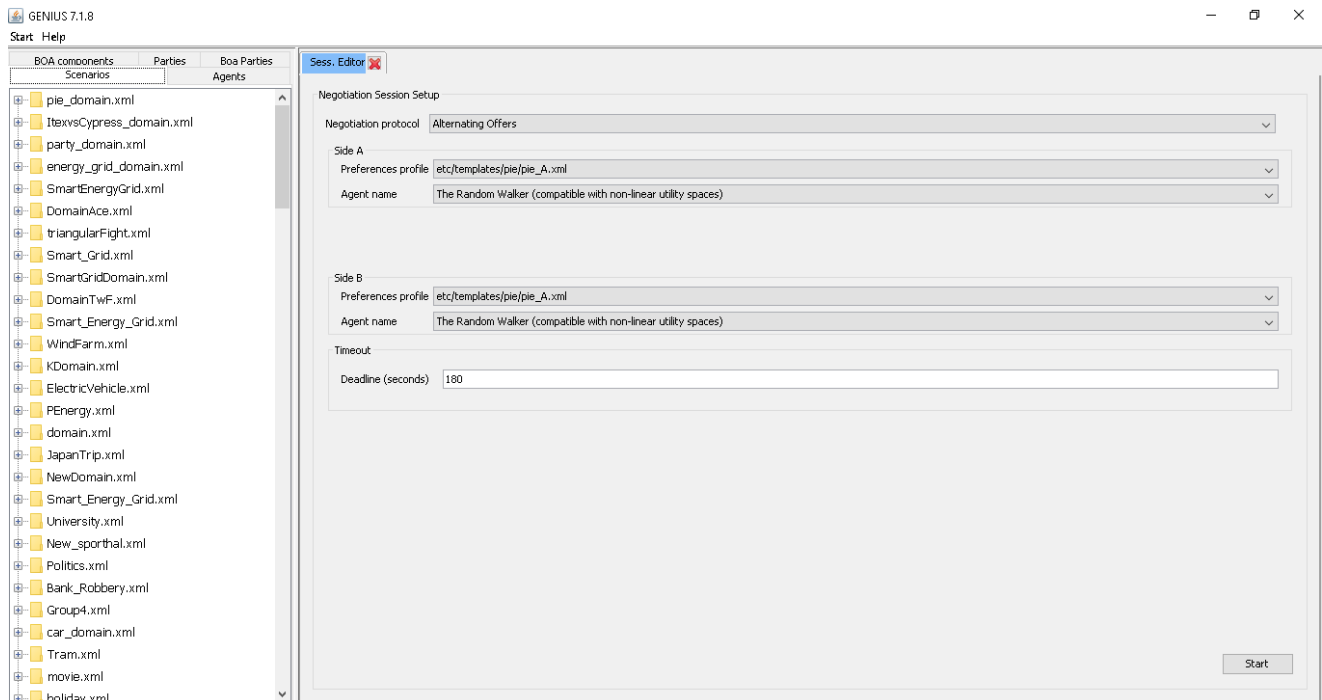


Figure 24: : Choisir les paramètres d'une négociation, ainsi que le protocole de négociation, deadline et les agents et leurs préférences, enfin, cliqué « start » pour commencer la négociation

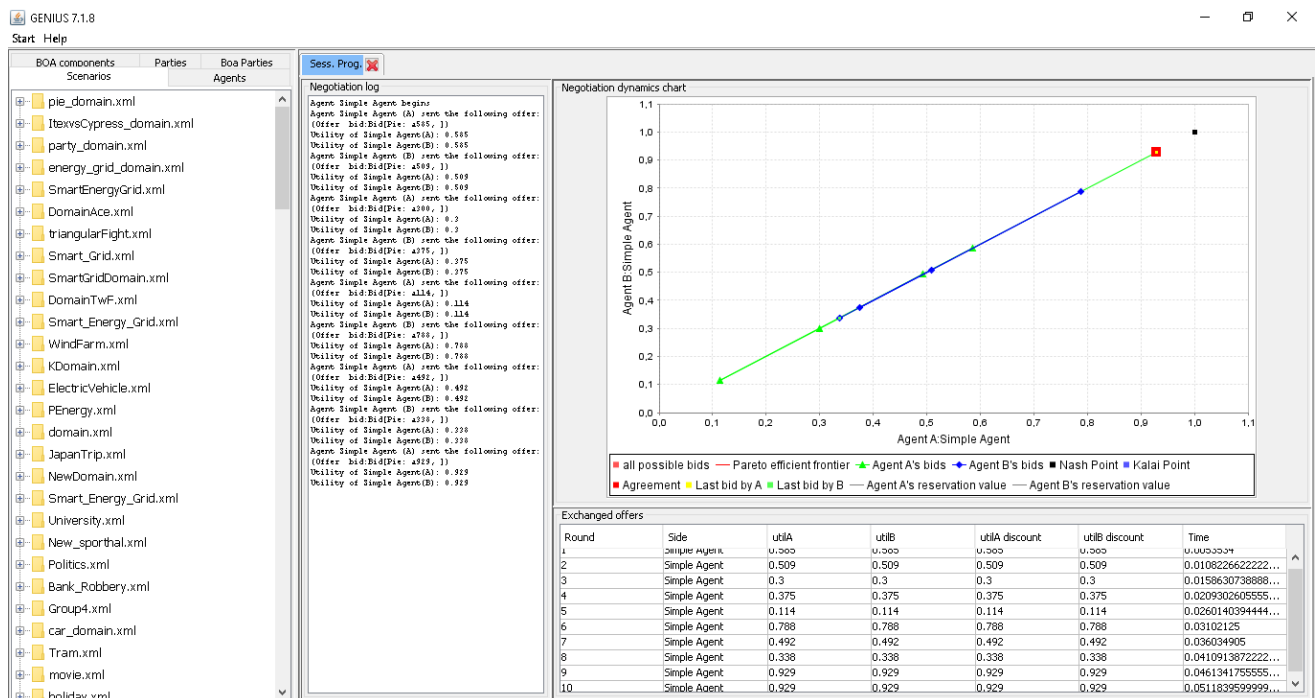


Figure 23 : Le résultat de la négociation.

3 La compétition des agents de négociation automatisés ANAC :

Le Concours International des Agents de Négociation Automatisés (ANAC) est un événement annuel qui vise à évaluer et à comparer les performances des agents de négociation automatisés dans des environnements de marché simulés. Il est organisé par la RoboCup Federation, une organisation internationale à but non lucratif qui promeut la recherche en robotique et en intelligence artificielle. Le concours ANAC est une plateforme qui réunit des chercheurs de la communauté de la négociation et fournit des repères uniques pour évaluer des stratégies de négociation pratiques dans des domaines à enjeux multiples. Les compétitions organisées ont favorisé l'émergence de nouvelles recherches en intelligence artificielle dans le domaine de la conception d'agents autonomes, et ont été mises à disposition de la communauté de recherche dans son ensemble. [66] La première édition de l'ANAC a eu lieu en 2005, et depuis lors, la compétition a été organisée chaque année. Le concours est ouvert aux chercheurs, aux étudiants et aux professionnels du domaine de l'IA qui souhaitent tester leurs algorithmes de négociation automatisés contre d'autres participants.

4 Eclipse :

Eclipse est un environnement de développement intégré (IDE) pour le développement d'applications utilisant le langage de programmation Java et d'autres langages de programmation tels que C/C ++, PHP, Python, PERL, Ruby etc. La plate-forme Eclipse, créée à l'aide de Java, est polyvalente et permet de développer une variété d'outils tels que des applications clients riches, des environnements de développement intégrés, etc. Grâce à la disponibilité de plug-ins, Eclipse peut être utilisé comme un environnement de développement intégré (IDE) pour pratiquement n'importe quel langage de programmation. [67] Grâce à Eclipse, il est possible de programmer des agents en établissant un lien de connexion entre Genius et Eclipse. Requier une machine virtuelle Java (JRE) pour marcher.

5 java (JDK) :

Java Development Kit (JDK) est un ensemble d'outils logiciels permettant aux développeurs de créer des applications Java. Il contient des compilateurs Java, des outils pour la documentation, des outils pour l'emballage et le déploiement de code, ainsi que d'autres outils utiles pour le développement de logiciels en Java. Le JDK comprend également la Java Runtime Environment (JRE), qui permet l'exécution d'applications Java sur une machine locale. Le JDK

est une plate-forme logicielle essentielle pour les développeurs Java, qui l'utilisent pour écrire, compiler, déboguer et déployer des applications Java.

6 Discussion des résultats :

6.1 Exemple d'application dans une session de négociation :

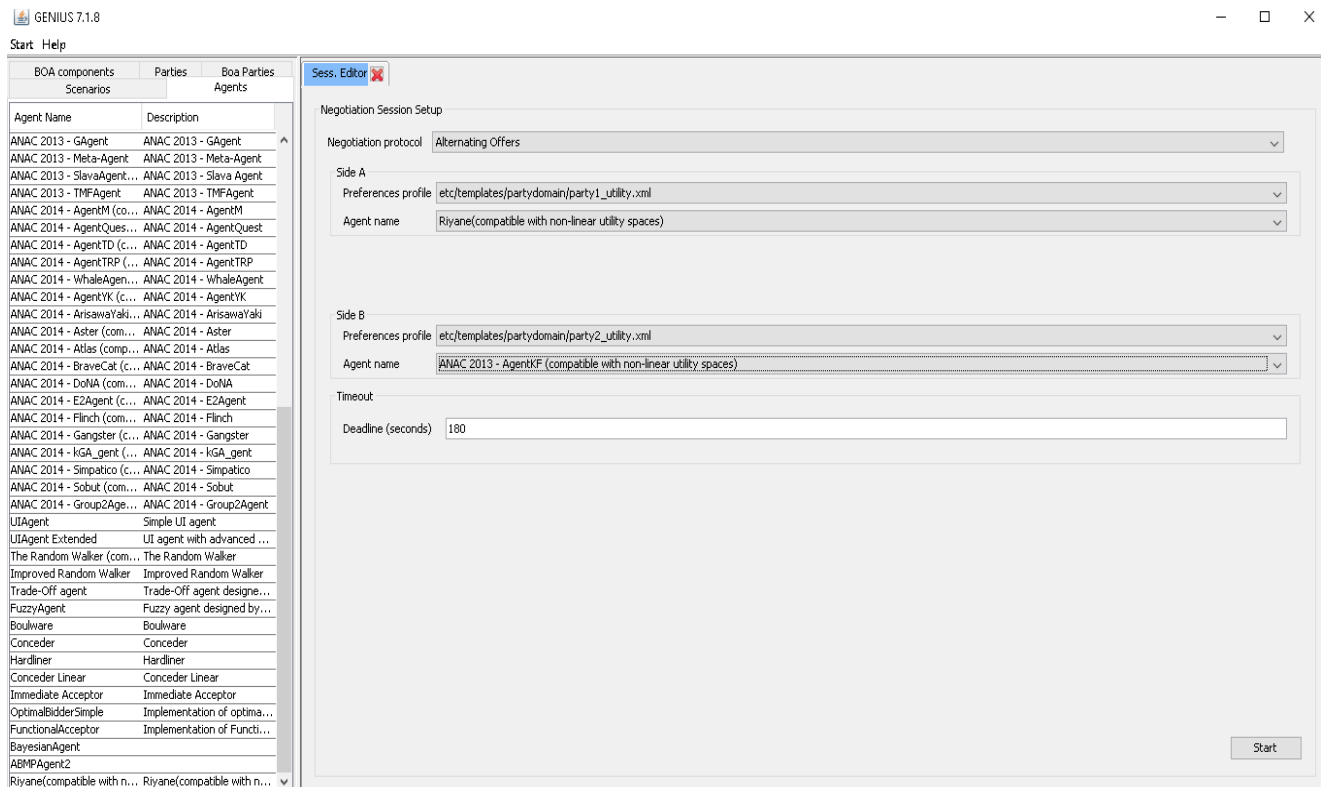


Figure 25: La session de négociation est selon le protocole alternative, notre agent

est de la coté « A » nommé« Riyane» avec l'agentKF

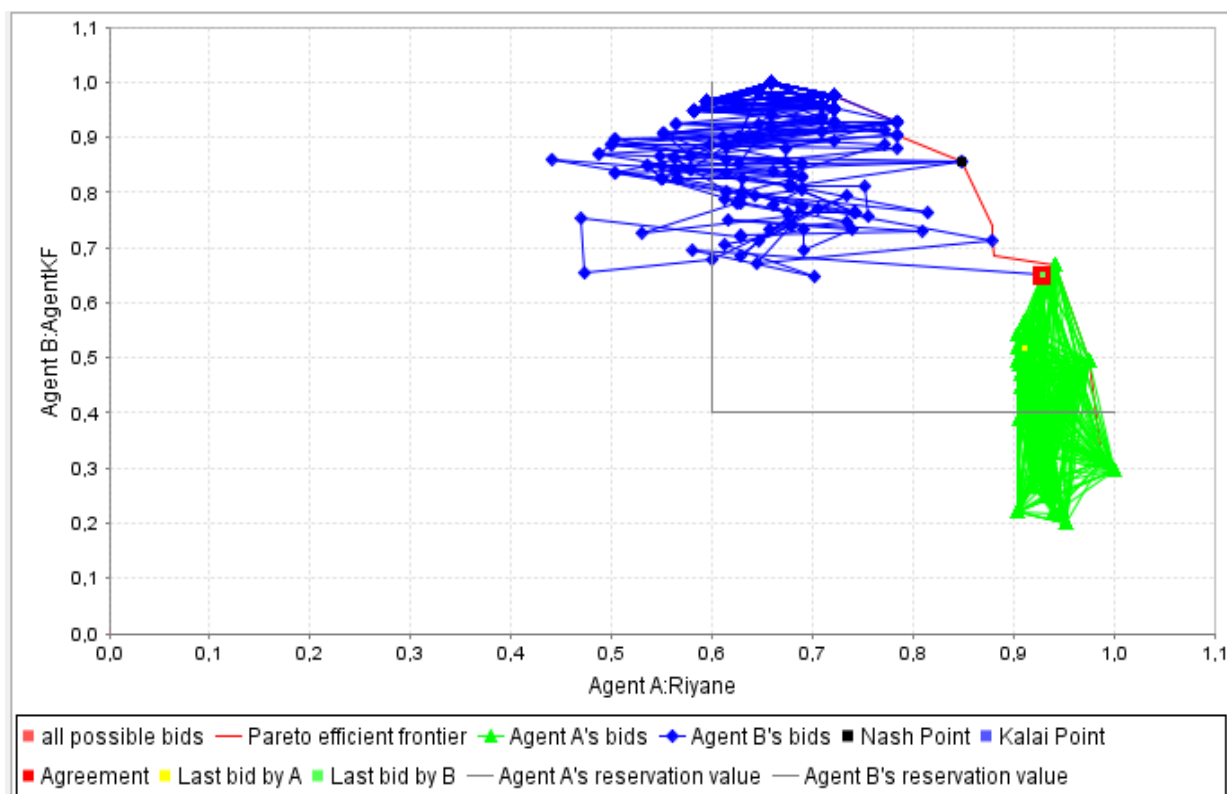


Figure 26: Les résultats selon la négociation session entre l'agent Riyane et l'agent KF .

Nous remarquons que notre agent est gagné la session avec une utilité max environ 0.929.

Les offres de notre agent au cours la négociation est comme suit :

Agent Riyane begins

Agent Riyane (A) sent the following offer:

(Offer bid:Bid[Food: Handmade Food, Drinks: Beer Only, Location: Party Room, Invitations: Custom, Handmade, Music: DJ, Cleanup: Water and Soap,])

Utility of Riyane(A): 0.9035557405886917

Utility of AgentKF(B): 0.5161726822666586

Agent AgentKF (B) sent the following offer:

(Offer bid:Bid[Food: Catering, Drinks: Handmade Cocktails, Location: Party Room, Invitations: Custom, Printed, Music: MP3, Cleanup: Water and Soap,])

Utility of Riyane(A): 0.6586134337130735

Utility of AgentKF(B): 1.0

Agent Riyane (A) sent the following offer:

(Offer bid:Bid[Food: Chips and Nuts, Drinks: Beer Only, Location: Party Room, Invitations: Custom, Handmade, Music: DJ, Cleanup: Water and Soap,])

Utility of Riyane(A): 0.9666591760120626

Utility of AgentKF(B): 0.46857814169718903

Agent AgentKF (B) sent the following offer:

(Offer bid:Bid[Food: Catering, Drinks: Handmade Cocktails, Location: Party Room, Invitations: Custom, Printed, Music: Band, Cleanup: Water and Soap,])

Utility of Riyane(A): 0.5952489329200307

Utility of AgentKF(B): 0.9639760602643335

Agent Riyane (A) sent the following offer:

.....

Utility of Riyane(A): 0.9291224619048221

Utility of AgentKF(B): 0.6498158026543664

6.2 Résultats obtenus :

Nous avons testé notre agent « Riyane » par tous les scénarios non linéaires disponibles dans GENIUS avec tous les agents compatibles avec les utilités non linéaires sous forme d'un tournoi.

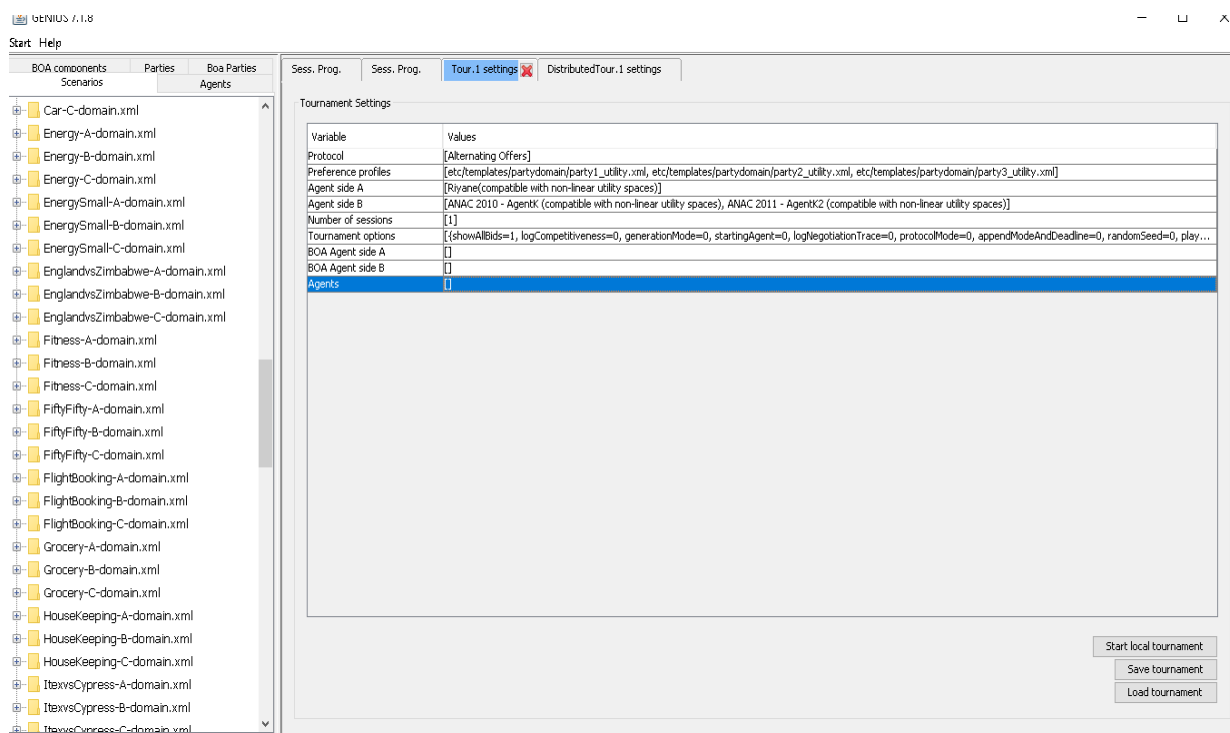


Figure 27: Une session tournoi avec tous les agents compatibles avec les utilités nonlinéaires sous GENIUS.

Après avoir cliqué sur « start local tournament », Le panneau est ensuite échangé pour une progression du tournoi.

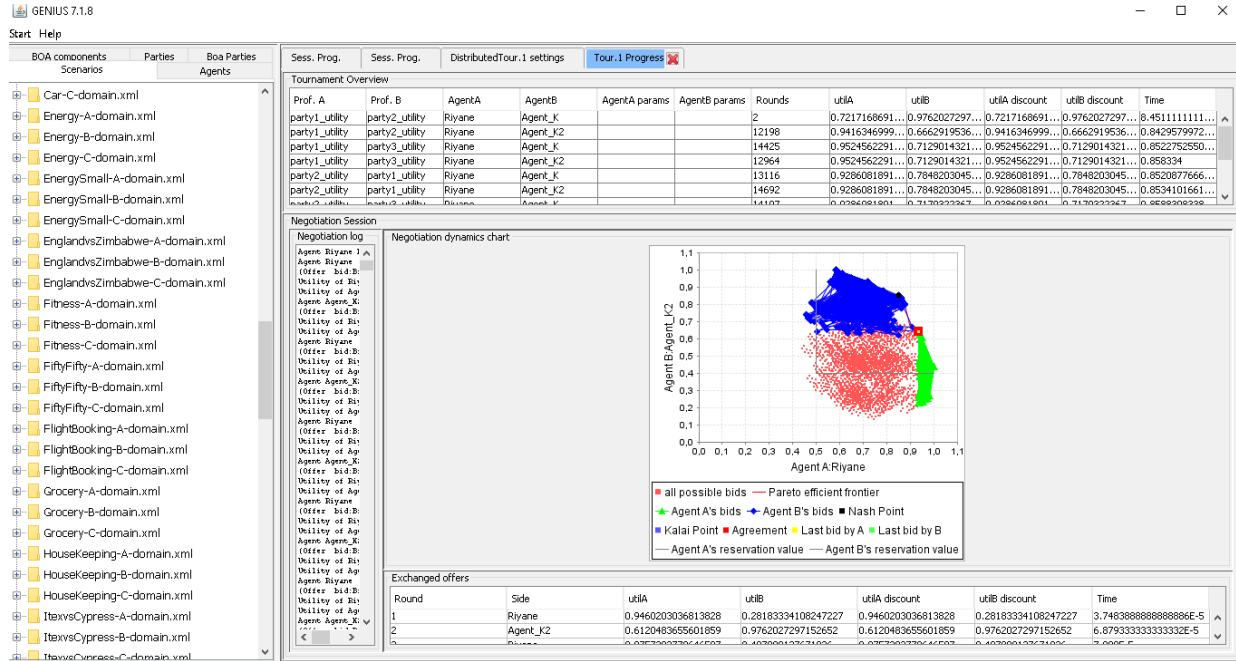


Figure 28: Les résultats du tournoi

Les tableaux suivants montrent tous les résultats de la session :

Prof. A	Prof. B	AgentA	AgentB	AgentA params	AgentB params	Rounds	utilA	utilB	utilA discount	utilB discount	Time
party1_utility	party2_utility	Riyane	Agent_K			2	0.7217168691...	0.9762027297...	0.7217168691...	0.9762027297...	8.4511111111...
party1_utility	party2_utility	Riyane	Agent_K2			12198	0.9416346999...	0.6662919536...	0.9416346999...	0.6662919536...	0.8429579972...
party1_utility	party3_utility	Riyane	Agent_K			14425	0.9524562291...	0.7129014321...	0.9524562291...	0.7129014321...	0.8522752550...
party1_utility	party3_utility	Riyane	Agent_K2			12964	0.9524562291...	0.7129014321...	0.9524562291...	0.7129014321...	0.858334
party2_utility	party1_utility	Riyane	Agent_K			13116	0.9286081891...	0.7848203045...	0.9286081891...	0.7848203045...	0.8520877666...
party2_utility	party1_utility	Riyane	Agent_K2			14692	0.9286081891...	0.7848203045...	0.9286081891...	0.7848203045...	0.8534101661...
party2_utility	party2_utility	Riyane	Agent_K			14107	0.9286081891...	0.7170222247...	0.9286081891...	0.7170222247...	0.9593020232...

Figure 29: Tableau de résultats

Prof. A	Prof. B	AgentA	AgentB	AgentA params	AgentB params	Rounds	utilA	utilB	utilA discount	utilB discount	Time
party2_utility	party3_utility	Riyane	Agent_K			14107	0.9286081891...	0.7179322367...	0.9286081891...	0.7179322367...	0.8588308338...
party2_utility	party3_utility	Riyane	Agent_K2			13258	0.9286081891...	0.7179322367...	0.9286081891...	0.7179322367...	0.8775243738...
party3_utility	party1_utility	Riyane	Agent_K			16322	0.9105738762...	0.7072442015...	0.9105738762...	0.7072442015...	0.85598671
party3_utility	party1_utility	Riyane	Agent_K2			18048	0.9285405625...	0.6738503159...	0.9285405625...	0.6738503159...	0.8610686327...
party3_utility	party2_utility	Riyane	Agent_K			16162	0.9358004045...	0.6396643296...	0.9358004045...	0.6396643296...	0.8398414188...
party3_utility	party2_utility	Riyane	Agent_K2			17110	0.9358004045...	0.6396643296...	0.9358004045...	0.6396643296...	0.8399678672...

Figure 30: Suite tableau de résultats.

Nous observons que notre agent termine le tournoi avec la plus grande utilité, finalement nous pouvons dire que notre agent « Riyane » capable d'apprendre leur opposant et maximiser leur valeur d'utilité.

7 Conclusion :

Au cours de ce chapitre, nous avons évalué les performances de notre agent "Riyane" dans la plateforme GENIUS en l'opposant à plusieurs autres agents capables de traiter des utilités non linéaires pour les paramètres d'apprentissage dans des négociations multi-issues non linéaires. Ces expérimentations nous ont permis de finaliser la conception d'un agent négociateur basé sur le Q-Learning.

Conclusion générale et perspectives

Dans ce mémoire, nous avons adopté une approche pour la conception de notre agent négociateur, qui consiste à agir de manière intelligente en apprenant les intentions de notre adversaire grâce à l'algorithme Q-learning, dans le but de maximiser notre utilité et de remporter les tours de négociation. Nous avons testé notre agent "riyane" sur la plateforme GENIUS, et nous avons obtenu des résultats satisfaisants tant pour notre agent que pour l'adversaire.

En fin de compte, ce travail nous encourage à envisager de tester notre agent sur différentes plateformes de simulation de négociations automatisées, en utilisant différentes stratégies d'apprentissage et en maximisant les objectifs, tels que la bibliothèque Negmas en Python. Nous souhaitons donc tester notre agent sur cette plateforme spécifique. Et nous pouvons changer l'algorithme Q-learning par l'algorithme SARSA ou Deep Learning ou apprentissage bayésien .

A. Références Bibliographiques

- [1] Pokhara, S. Schiffer, O. Obst, and K. Fischer, "A Brief Survey of Multi-Agent Systems," in *Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications*, H. Yu and J. Cao, Eds. London: InTech, 2010, pp. 1-18..
- [3] Gerhard Weiss. *Multiagent systems : a modern approach to distributed artificial intelligence*. MIT press, 1999...
- [4] Marie-France Blanquet. « Web collaboratif, web coopératif,..
- [5] Artikel 1745 Ger. W...
- [6] N. PORTUGAELS, *De vaststellingsovereenkomst*, Antwerpen, Intersentia, 2018, 200. Voor dading: artt. 2044 tot 2058 BW.
- [8] Idem, p. 123..
- [10] "Systèmes multi-agents : Vers une intelligence collective" de Jacques Ferber, publié en 1995 aux éditions InterEditions..
- [12] (BENAZI Makhoulouf, *Négociation des offres dans les enchères par les systèmes multi agents* , MEMOIRE , UNIVERSITE FERHAT ABBAS – SETIF , 2007 ,37).
- [14] B. An, N. Gatti, and V. Lesser. Alternating-offers bargaining in one-to-many and many-to-many settings. *AMAI*, 77(1-2) :67–103, 2016..
- [15] J. S. Rosenschein and G. Zlotkin. *Rules of encounter : designing conventions for automated negotiation among computers*. MIT press, 1994..
- [16] Lax, D. A., & Sebenius, J. K. (1986). *The manager as negotiator: Bargaining for cooperation and competitive gain*. Free Press..
- [17] Lewicki, R. J., Saunders, D. M., & Barry, B. (2010). *Negotiation*. McGraw Hill..
- [18] Alpaydin, E. (2010). *Introduction to machine learning*. MIT Press..
- [19] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press..
- [20] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer..
- [21] Domingos, P. (2015). *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.
- [22] Kelleher, J. D., & Tierney, B. (2018). *Data science: An introduction*. CRC Press..
- [23] "Pattern Recognition and Machine Learning" de Christopher M. Bishop..
- [24] "Machine Learning: A Probabilistic Perspective" de Kevin P. Murphy.
- [25] "Hands-On Machine Learning with Scikit-Learn and TensorFlow" d'Aurélien Géron..

- [26] Alpaydin, E. (2010). Introduction to machine learning (2nd ed.). Cambridge, MA: MIT Press
- [27] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer..
- [28] "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" de Aurélien Géron..
- [29] "Machine Learning Approaches to Spam Filtering" par G. Cormack et al. (2007)..
- [30] "A Survey on Voicebot: Data Set, Techniques and Evaluation Metrics" par N. Khatun et al. (2021)..
- [31] "A Survey of Autonomous Vehicle Technology" par D. S. Levinson et al. (2011)..
- [36] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer..
- [40] Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 036106..
- [41] Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (CSUR)*, 45(4), 43.
- [42] Xie, J., Szymanski, B. K. (2013). LabelRank: A Stabilized Label Propagation Algorithm for Community Detection in Networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(11), 2716-2729..
- [44] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
- [50] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780
- [52] .H. Ahmed et al. / Deep Reinforcement Learning for Multi-Agent Interaction
- [53] Medet Kanmaz and Elif Surer/ Using Multi-Agent Reinforcement Learning in Auction Simulations
- [55] Zerrougui Amani/Elicitation des préférences dans un domaine de négociation multi-issues non linéaire/p52
- [56] Yoav Shoham, Rob Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007.
- [57] Karl Tuyls and Gerhard Weiss. Multiagent learning: Basics, challenges, and prospects. *AI Magazine*, 33(3):41–52, 2012. [7] Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3), November 2005.
- [60] «MLQ, deep reinforcement learning guide to deep Q learning,» [En ligne]. Available: <https://www.mlq.ai/deep-reinforcement-learning-q-learning/>. [Accès le 21 06 2020].
- [61] DILMI Abdallah./ APPRENTISSAGE PAR RENFORCEMENT EN UTILISANT LES RESEAUX DE NEURONES ARTIFICIELS: SIMULATION DE LA NAVIGATION D'UN ROBOT MOBILE/p15

[62] Anthony JUTON – Valentin NOËL – Rida LALI/Introduction à l'apprentissage par renforcement/p10

[63] A. Florea, «agents intelligent cour web interactif, chapitre 2: architecture des agents et langage,» Politechnia University of Bucharest, 2002

[64] Zerrougui Amani/Elicitation des préférences dans un domaine de négociation multi-issues non linéaire/p52

[65] S. K. e. a. RAZ LIN, «G E N I U S: AN INTEGRATED ENVIRONMENT FOR SUPPORTING THE DESIGN OF GENERIC AUTOMATED NEGOTIATORS,» computational intelligent, p. 23, 2012.

[66] Zerrougui Amani/Elicitation des préférences dans un domaine de négociation multi-issues non linéaire/p72

po

B. Références Web

[2] <https://economy-pedia.com/11040461-distributive-negotiation#menu-1..>

[7] <https://members.irglobal.com/file/4aae0f1bcfd7a40bbe06cb4706dda149.pdf>.

[9] <https://celinevallieres.com/la-negotiation-competitive-pas-si-efficace/>.

[11] <https://turing.cs.pub.ro/auf2/html/chapters/chapter5/chapitre-5.pdf>.

[13] <https://ceur-ws.org/Vol-2133/cnia-paper1.pdf>.

[32] [https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501909-regression-logistique-definition-et-cas-d-usage-en-ia/..](https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501909-regression-logistique-definition-et-cas-d-usage-en-ia/)

[33] https://www.google.com/url?sa=i&url=https%3A%2F%2Faws.amazon.com%2Ffr%2Fwhat-is%2Flogistic-regression%2F&psig=AOvVaw0N5zVFeTYlucl_qmIgiwWd&ust=1678275641179000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCPCb0Y3eyf0CFQAAAAAdAAAAABAE..

[34] [https://dataanalyticspost.com/Lexique/svm/#:~:text=SVM%20\(Support%20Vector%20Machine%20ou,ou%20de%20d%C3%A9tection%20d'anomalie..](https://dataanalyticspost.com/Lexique/svm/#:~:text=SVM%20(Support%20Vector%20Machine%20ou,ou%20de%20d%C3%A9tection%20d'anomalie..)

[35] https://dataanalyticspost.com/wp-content/uploads/2018/01/img_svm.png.

[37] <https://hastie.su.domains/Papers/ESLII.pdf>.

[38] <https://www.vldb.org/conf/1994/P487.PDF>.

[39] http://users.eecs.northwestern.edu/~yingliu/datamining_papers/survey.pdf.

[43] https://projets-info.insa-rennes.fr/projets/2016/La_Belle_Propagation/forcedirected.png.

[45] <https://ryax.tech/fr/deep-learning-comprendre-les-reseaux-de-neurones-artificiels-artificial-neural-networks/>

[46] <https://www.ionos.fr/digitalguide/fileadmin/DigitalGuide/Schaubilder/FR-reseau-de-neurones-artificiels.png>

- [47] <https://images.squarespacecdn.com/content/v1/519a7bc0e4b08ccdf8f31445/1523974153534-9PXNUUHYFWH9MY0R6THD/conv.png>
- [48] <https://datasciencetoday.net/index.php/en-us/deep-learning/173-les-reseaux-de-neurones-convolutifs>
- [49] <https://cs231n.github.io/assets/cnn/convnet.jpeg>
- [51] <https://www.researchgate.net/publication/343135807/figure/fig5/AS:916109729492996@1595428988218/Reseau-de-neurones-recurrents.ppm>
- [54] <https://www.researchgate.net/publication/340475664/figure/fig1/AS:877526758981632@1586230091147/A-reinforcement-learning-mechanism-for-an-auction-design.jpg>
- [58] <https://www.jedha.co/formation-ia/algorithmes-q-learning#:~:text=Qu'est%20ce%20que%20le%20Q%20learning%20%3F,dans%20un%20contexte%20bien%20pr%C3%A9cis.>
- [59] <https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/>
- [67] <https://www.youtube.com/watch?v=Ax8B20Ipho>

Genius est un environnement qui met en œuvre une architecture ouverte pour des agents de négociation hétérogènes. Genius peut être utilisé pour mettre en œuvre ou simuler des négociations de la vie réelle. Et pour mener une négociation, il existe plusieurs modes pour exécuter une négociation parmi eux :

1. **Session de négociation** : Une session de négociation concerne une négociation unique dans laquelle deux agents s'affrontent. Ce mode est principalement destiné aux nouveaux utilisateurs.

Pour exécuter une session de négociation, sélectionnez "start", puis "Negotiation Session". Les paramètres suivants doivent être spécifiés pour exécuter une négociation :

- **Negotiation protocol** : l'ensemble des protocoles disponibles. Normalement, on utilise "Alternating Offers".
 - **Side A/Side B** : la configuration des agents des deux parties.
 - **Preference profile** : le profil de préférence à utiliser par l'agent de cette partie.
 - **Agent name** : l'agent participant à la négociation.
 - **Deadline (secondes)** : la durée de la négociation en secondes.
2. **Tournoi** : Un tournoi est un ensemble de sessions. Deux ensembles d'agents s'affrontent sur un ensemble de domaines. Les résultats des sessions sont stockés dans le répertoire « log ». En plus d'exécuter une session de négociation individuelle, il est également possible d'organiser un tournoi. Un tournoi peut être considéré comme une collection de sessions. Contrairement à l'exécution d'une session unique, les résultats d'un tournoi sont enregistrés dans le répertoire "log". Un tournoi peut être créé en sélectionnant d'abord "start", puis "Tournament". Cette fenêtre affiche un ensemble d'options que nous devons spécifier :
 - **Protocol** : L'ensemble des protocoles disponibles.
 - **Preference profiles** : L'ensemble des scénarios sur lesquels les agents doivent concourir. Chaque scénario sélectionné doit comporter au moins deux profils de préférence.
 - **Agent side A/B** : L'ensemble des agents du côté A qui concourent contre tous les agents du côté B.
 - **Number of sessions** : Le nombre de fois que chaque session doit être répétée.
 - **Tournament options** : Les options qui spécifient comment exécuter le tournoi.
 - **BOA Agent side A/B** : Type d'agents composés de plusieurs composants.

