



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene



Faculté d'informatique

Module : Métaheuristique et algorithmes évolutionnaires

Projet Tech Agent

Binôme

LOUNAS Katia

191931012680

SADI OUFELLA Ouiza

191931012441

30/03/2023

Groupe 04

Table des matières

Table des matières.....	1
Introduction.....	2
Partie 01 (Système expert).....	3
Système expert.....	3
Chaînage avant.....	3
Implémentation.....	4
Base de faits.....	4
Base de règle.....	5
Gestion de conflit.....	6
Interface.....	6
Partie 02 (Enchère).....	9
Problème de détermination de gagnant.....	9
Système multi Agent.....	10
Implémentation.....	10
Agent vendeur.....	10
Agent acheteur.....	11
Résultat.....	11
Conclusion.....	13

Introduction

Les agents intelligents se réfèrent à des programmes informatiques qui démontrent la capacité de percevoir leur environnement, de prendre des décisions indépendantes et de répondre en conséquence afin d'accomplir des objectifs précis. Leur conception vise à simuler diverses aptitudes cognitives et comportementales humaines, telles que la perception, la réflexion, l'apprentissage et les interactions sociales. Ces agents utilisent des méthodes d'intelligence artificielle, telles que l'apprentissage automatique, la logique et la planification, pour acquérir des connaissances, résoudre des problèmes et s'adapter aux situations changeantes. Ils sont appliqués dans de multiples domaines, notamment la robotique, les systèmes de recommandation, la gestion des ressources, la surveillance et les jeux, entre autres. Les agents intelligents peuvent opérer de manière autonome ou collaborer avec d'autres agents en vue d'atteindre des objectifs communs. Leur conception vise à améliorer l'efficacité, l'adaptabilité et les capacités décisionnelles dans des contextes complexes et dynamiques.

Partie 01 (Système expert)

Système expert

Un système expert est un système informatique qui utilise des connaissances expertes pour résoudre des problèmes complexes dans un domaine spécifique. Il s'agit d'un type de système d'IA qui utilise des règles logiques et des connaissances expertes pour simuler le raisonnement humain et prendre des décisions en fonction des informations fournies.

Un système expert est composé de deux parties principales : la base de connaissances et le moteur d'inférence. La base de connaissances est une collection de faits, de règles et de heuristiques qui représentent les connaissances d'experts dans un domaine particulier. Le moteur d'inférence est responsable de l'application de ces connaissances pour résoudre un problème spécifique.

Le système expert fonctionne en posant des questions à l'utilisateur pour collecter des informations sur le problème. Ces informations sont ensuite utilisées pour rechercher des faits et des règles dans la base de connaissances, afin de prendre des décisions ou de fournir des recommandations. Le système peut également apprendre de ses erreurs et s'améliorer avec le temps, en recueillant des commentaires et en ajustant les règles et les connaissances dans la base de données.

Les systèmes experts ont été largement utilisés dans des domaines tels que la médecine, la finance, l'ingénierie, la sécurité, etc. Ils permettent aux utilisateurs de résoudre des problèmes complexes en fournissant des réponses rapides et précises en utilisant l'expertise et les connaissances d'experts.

Chaînage avant

Le chaînage avant constitue une méthodologie d'inférence employée dans le domaine de l'intelligence artificielle pour parvenir à des déductions supplémentaires à partir d'un ensemble de règles et de faits de départ.

Cette procédure s'amorce en examinant les faits initiaux à notre disposition. Par le biais de l'application des règles de déduction, le système s'efforce de parvenir à de nouvelles informations en se basant sur les faits déjà établis. Dans cette optique, chaque règle est examinée une par une afin de déterminer si les conditions associées concordent avec les faits actuels. Lorsqu'une règle dont les conditions sont

remplies est identifiée, elle est activée, et une nouvelle information est ainsi déduite. Celle-ci est ensuite incorporée à la base de faits. Ce processus se répète jusqu'à ce qu'il n'y ait plus de nouvelles informations à inférer. Le chaînage avant est un processus itératif qui scrute de façon séquentielle les règles et les faits. Il se prête à la résolution de problèmes de diagnostic, de planification, de prise de décision, et bien d'autres encore. En fin de compte, le résultat final se traduit par un ensemble de faits déduits, qui peuvent être exploités pour prendre des décisions ou entreprendre des actions appropriées en conséquence.

Implémentation

Dans cette partie nous avons développé un système expert dans les régimes alimentaires.

Base de faits

Notre base contient en total 9 faits avec une variation de valeurs.

Fait	valeur
Taille	Variable
Âge	variable
Poids	variable
Sexe	Femme,Homme
Activité physique	Niveau Sédentaire, Niveau Léger, Niveau Modéré, Niveau Actif, Niveau Très actif
Condition médicale	Diabète de type 2, Hypertension artérielle, Syndrome de l'intestin irritable
Préférences alimentaires	Végétarien, Carnivore, Végétalien
Allergies alimentaires	Allergie aux arachides, Allergie au gluten, Allergie au lait
Objectif	Maintien du poids, Prise de masse musculaire, Perte de poids

Base de règle

Dans notre système expert, nous avons développé un ensemble de 21 règles qui constituent sa base de connaissances. Chaque règle a été attribuée une priorité spécifique, et une fois qu'une règle a été utilisée, elle est automatiquement désactivée. Cela permet d'assurer un fonctionnement efficace et une prise de décision précise dans le système expert.

```
// Règles basées sur les allergies alimentaires
regles.add(new Regle(Arrays.asList("Allergie aux arachides", "Femme",
"Adulte", "Tres leger"), "Régime sans arachides", true, 1));
regles.add(new Regle(Arrays.asList("Allergie au gluten", "Homme", "Normal",
"Adulte"), "Régime sans gluten", true, 2));
regles.add(new Regle(Arrays.asList("Allergie au lait", "Normal", "Adulte"),
"Régime sans lactose", true, 3));
regles.add(new Regle(Arrays.asList("Homme", "Allergie aux arachides",
"Normal", "Adulte"), "Régime sans arachides", true, 4));

regles.add(new Regle(Arrays.asList("Allergie aux arachides"), "Régime sans
arachides", true, 19));
regles.add(new Regle(Arrays.asList("Allergie au gluten"), "Régime sans gluten",
true, 20));
regles.add(new Regle(Arrays.asList("Allergie au lait"), "Régime sans lactose",
true, 21));

// Règle basée sur le niveau d'activité physique, les préférences alimentaires et
les conditions médicales
regles.add(new Regle(Arrays.asList("Niveau Modéré", "Végétarien", "Syndrome
de l'intestin irritable"), "Régime végétarien", true, 5));

// Règles basées sur les conditions médicales
regles.add(new Regle(Arrays.asList("Diabète de type 2", "Normal"), "Réduction
du sucre", true, 6));
regles.add(new Regle(Arrays.asList("Hypertension artérielle", "Lourd",
"Adulte"), "Régime pour l'hypertension", true, 7));
regles.add(new Regle(Arrays.asList("Syndrome de l'intestin irritable", "Adulte",
"Normal"), "Réduction du cholestérol", true, 8));

// Règles basées sur le niveau d'activité physique
regles.add(new Regle(Arrays.asList("Niveau Sédentaire", "Enfant", "Lourd"),
"Perte de poids", true, 9));
regles.add(new Regle(Arrays.asList("Niveau Léger", "Femme", "Adulte",
"Lourd"), "Perte de poids", true, 10));
regles.add(new Regle(Arrays.asList("Niveau Modéré", "Homme", "Adulte",
"Allergie au gluten"), "Régime sans gluten", true, 11));
regles.add(new Regle(Arrays.asList("Actif", "Adulte", "Tres leger"), "Prise de
masse musculaire", true, 12));
regles.add(new Regle(Arrays.asList("Très actif", "Adulte", "Tres leger"), "Prise de
masse musculaire", true, 13));
// Règles basées sur les préférences alimentaires
regles.add(new Regle(Arrays.asList("Végétarien", "Normal", "Adulte"), "Régime
```

```

végétarien", true, 14));
regles.add(new Regle(Arrays.asList("Carnivore", "Normal", "Adulte"), "Perte de
poids", true, 15));
regles.add(new Regle(Arrays.asList("Végétalien", "Adulte", "Tres leger"), "Prise
de masse musculaire", true, 16));

// Règles basées sur le sexe
regles.add(new Regle(Arrays.asList("Femme", "Adulte", "Tres leger"), "Prise de
masse musculaire", true, 17));

// Règle basée sur l'objectif nutritionnel, l'âge, le poids, le sexe et les allergies
alimentaires
regles.add(new Regle(Arrays.asList("Perte de poids", "Adulte", "Normal",
"Femme", "Allergie au gluten"), "Maintien du poids", true, 18));

```

Gestion de conflit

Pour résoudre les conflits entre les règles à appliquer, nous avons mis en place une fonction qui gère les priorités en fonction du nombre de prémisses et si ce dernier avère insuffisant on procède selon l'ordre de création de chaque règle. Si une règle a un plus grand nombre de prémisses, elle se voit attribuer la plus haute priorité. Cette approche nous permet de gérer efficacement les conflits et de déterminer quelle règle doit être appliquée en cas de concurrence entre plusieurs règles.

Interface

Dans ce cas, notre système expert parcourt tous les objectifs sans en spécifier un particulier, afin d'y attribuer un résultat cohérent.

The screenshot shows a web application window titled "Healtho". The main heading is "RÉGIME ALIMENTAIRE" with the subtitle "Utilisez des ingrédients frais et saints." Below this is a form titled "Objectif" with an "Optionnel" label and a green "Start" button. The form contains several input fields:

- Age:** 25
- Sexe:** Femme (dropdown menu)
- Activité physique:** Niveau Léger (dropdown menu)
- Alergies alimentaire:** Allergie aux arachi... (dropdown menu)
- Taille:** 165
- Poids:** 55
- Préférences:** Végétarien (dropdown menu)
- Conditions médicales:** Diabète de type 2 (dropdown menu)

Below the form, a dark grey box displays the following text:

Le résultat de notre diagnostique montre que il vous faut un ' Régime sans arachides '

Assurez-vous de prendre soin de vous en adoptant une alimentation équilibrée N'hésiter pas de consulter un médecin et suivre ses recommandations

Merci pour votre confiance

Cas au le but spécifié ne correspond pas aux besoins individuels

Healtho

RÉGIME ALIMANTAIRE

Utilisez des ingrédients frais et saints.

Objectif
Régime sans gluten Start

Age
25

Sexe
Femme

Activité physique

Alergies alimentaire
Allergie au gluten

Taille
165

Poids
56

Préférences

Conditions médicales

Le résultat de notre diagnostic montre que il vous faut un ' Régime sans gluten '.

Assurez-vous de prendre soin de vous en adoptant une alimentation équilibrée N'hésiter pas de consulter un médecin et suivre ses recommandations.

Merci pour votre confiance

Cas avec un but approprié à la personne.

Healtho

RÉGIME ALIMANTAIRE

Utilisez des ingrédients frais et saints.

Objectif
Perte de poids Start

Age
19

Sexe
Femme

Activité physique

Alergies alimentaire
Allergie au gluten

Taille
165

Poids
40

Préférences

Conditions médicales

Nous vous conseillons d'être attentif lors du choix d'un régime, car c'est une décision délicate qui nécessite une réflexion approfondie.

Le régime actuel ne semble pas adapté à votre situation. Afin de vous fournir un régime plus approprié, nous vous recommandons d'utiliser notre système sans spécifier d'objectif particulier.

Cela nous permettra d'évaluer votre situation de manière plus globale et de vous proposer un régime plus adapté à vos besoins spécifiques.

Merci pour votre confiance

Partie 02 (Enchère)

Problème de détermination de gagnant

Les enchères sont un mécanisme couramment utilisé pour déterminer le gagnant lors de la vente ou de l'achat de biens ou services. Cependant, la détermination du gagnant peut parfois poser problème.

Dans certaines enchères, le gagnant est simplement la personne qui fait l'offre la plus élevée. Cependant, cela peut entraîner des inconvénients tels qu'une surenchère excessive, où les participants sont incités à proposer des prix démesurés pour remporter l'enchère.

Pour remédier à cela, différentes méthodes ont été développées. Par exemple, certaines enchères utilisent des prix de réserve, où le vendeur fixe un prix minimum qu'il est prêt à accepter. Si aucune offre n'atteint ce prix, l'enchère peut être annulée ou le bien peut être remis en vente.

Une autre approche consiste à utiliser des enchères à prix scellé, où les participants soumettent leurs offres de manière confidentielle. Dans ce cas, le gagnant est celui qui a fait l'offre la plus élevée, mais le prix final payé est généralement celui de la deuxième offre la plus élevée. Cela encourage les participants à proposer un prix réaliste et éviter une surenchère excessive.

Il existe également des enchères dites "dynamiques" ou "ascendantes-descendantes" où le prix est ajusté en fonction des offres des participants. Cela permet d'atteindre un équilibre entre le prix le plus élevé que les participants sont prêts à payer et la valeur réelle du bien ou du service.

En résumé, le problème de la détermination du gagnant dans les enchères est abordé de différentes manières afin d'assurer un processus équitable et de maximiser la valeur pour toutes les parties impliquées.

Dans cette section, nous nous pencherons sur la résolution de cette problématique en utilisant la technologie des agents et en exploitant la plateforme Jade dans le contexte des enchères anglaises.

Système multi Agent

Un système multi-agent est constitué de plusieurs entités autonomes, désignées sous le nom d'agents, qui interagissent et collaborent en vue d'atteindre un objectif commun. Chacun de ces agents dispose de ses propres connaissances, capacités de perception et d'action, et peut prendre des décisions de manière indépendante. Les agents peuvent échanger des informations, partager des ressources et coordonner leurs actions afin de résoudre des problèmes complexes. Les systèmes multi-agents sont appliqués dans divers domaines tels que la robotique, l'intelligence artificielle, les réseaux sociaux et la gestion des ressources. Ils permettent une répartition des tâches, une résolution de problèmes distribuée et une adaptation en temps réel, ce qui leur confère une grande souplesse et adaptabilité. L'interaction entre les agents peut se dérouler de manière coopérative ou compétitive, en fonction du contexte et des objectifs du système.

Implémentation

Dans le cadre de cette section, nous avons procédé à la mise en place de deux agents distincts, à savoir un agent vendeur et plusieurs agents acheteurs.

Agent vendeur

L'agent vendeur est une entité qui gère le déroulement de l'enchère et communique avec les agents acheteurs. Voici une explication du code :

La classe "Vendeur" est déclarée en tant qu'extension de la classe "Agent" fournie par la plateforme JADE.

Différentes variables sont déclarées pour gérer les informations relatives à l'enchère, telles que la meilleure offre, l'offre en cours, le tour actuel, le nombre d'acheteurs, etc.

La méthode "**setup()**" est utilisée pour initialiser l'agent vendeur et définir ses comportements.

Un comportement cyclique est ajouté pour gérer les réponses des acheteurs. L'agent vendeur attend les messages de type **PROPOSE** (proposition d'offre) et vérifie si l'offre est supérieure à la meilleure offre actuelle. Si c'est le cas, il met à jour la meilleure offre et enregistre l'acheteur correspondant.

Un comportement à intervalle régulier est ajouté pour démarrer un nouveau tour d'enchères. À chaque intervalle défini (10 000 unités de temps), le vendeur prépare

un nouveau tour en incrémentant l'offre actuelle et en envoyant des messages de type **CFP** (appel d'offres) à tous les acheteurs.

Un comportement à intervalle régulier est également ajouté pour mettre fin à l'enchère après un certain nombre de tours. À chaque intervalle défini (50 000 unités de temps), le vendeur affiche un message indiquant la fin de l'enchère, le gagnant et l'offre remportée, puis l'agent vendeur est supprimé.

La méthode "**takeDown()**" est utilisée pour gérer la terminaison de l'agent vendeur et afficher un message indiquant sa fermeture.

Agent acheteur

L'agent acheteur est une entité qui participe à l'enchère en soumettant des offres au vendeur. Voici une explication du code :

La classe Acheteur est déclarée en tant qu'extension de la classe "Agent" fournie par la plateforme **JADE**.

Différentes variables sont déclarées pour gérer les informations de l'acheteur, telles que le solde (balance) disponible et l'incrément d'enchères (bid Increment).

La méthode "**setup()**" est utilisée pour initialiser l'agent acheteur et définir son comportement.

Un comportement **cyclique** est ajouté pour gérer les demandes de l'agent vendeur. L'agent acheteur attend les messages de type **CFP** (appel d'offres) et vérifie s'il peut faire une offre en fonction de son solde disponible. Si le solde est suffisant, il crée un message de réponse de type **PROPOSE** contenant une offre supérieure à l'offre actuelle et l'envoie au vendeur. Sinon, il affiche un message indiquant qu'il ne peut plus faire d'offre.

Le comportement cyclique permet à l'agent acheteur de continuer à attendre de nouvelles demandes de l'agent vendeur.

En résumé, ce code implémente un agent acheteur qui participe à l'enchère en répondant aux demandes de l'agent vendeur avec des offres supérieures à l'offre actuelle. L'agent acheteur vérifie également son solde disponible avant de faire une offre.

Résultat

Début de l'enchère, chaque agent acheteur propose son prix.

```
Un Agent Bidder Acheteur2@192.168.43.51:1099/JADE vient d'arrivé.
Un Agent Bidder Acheteur1@192.168.43.51:1099/JADE vient d'arrivé.
Agent Auctioneer Vendeur ouvre l'enchère.
-----
Auctioneer Agent: Enchène avec le tour 1
Bidder Agent: propose une offre de 9 pour l'Agent Auctioneer
Bidder Agent: propose une offre de 9 pour l'Agent Auctioneer
Auctioneer Agent: La nouvelle meilleure offre est 9 proposer par Acheteur1
-----
Auctioneer Agent: Enchène avec le tour 2
Bidder Agent: propose une offre de 16 pour l'Agent Auctioneer
Bidder Agent: propose une offre de 16 pour l'Agent Auctioneer
Auctioneer Agent: La nouvelle meilleure offre est 16 proposer par Acheteur1
-----
Auctioneer Agent: Enchène avec le tour 3
Bidder Agent: propose une offre de 29 pour l'Agent Auctioneer
Bidder Agent: propose une offre de 31 pour l'Agent Auctioneer
Auctioneer Agent: La nouvelle meilleure offre est 29 proposer par Acheteur1
Auctioneer Agent: La nouvelle meilleure offre est 31 proposer par Acheteur2
-----
```

L'enchère s'est conclue avec la victoire de l'acheteur numéro 1

```
-----
Auctioneer Agent: Enchène avec le tour 3
Bidder Agent: propose une offre de 29 pour l'Agent Auctioneer
Bidder Agent: propose une offre de 31 pour l'Agent Auctioneer
Auctioneer Agent: La nouvelle meilleure offre est 29 proposer par Acheteur1
Auctioneer Agent: La nouvelle meilleure offre est 31 proposer par Acheteur2
-----
Auctioneer Agent: Enchène avec le tour 4
Bidder Agent: propose une offre de 42 pour l'Agent Auctioneer
Bidder Agent: propose une offre de 44 pour l'Agent Auctioneer
Auctioneer Agent: La nouvelle meilleure offre est 42 proposer par Acheteur2
Auctioneer Agent: La nouvelle meilleure offre est 44 proposer par Acheteur1
Auctioneer Agent: Fin de l'enchère.
l'objet est remporté par l'Acheteur1 avec l'offre de 44
Auctioneer Agent Vendeur has terminated.
```

Conclusion

Ce projet nous a octroyé une occasion d'appréhender l'importance et le fonctionnement des systèmes experts ainsi que des systèmes multi-agents. Nous avons pu observer avec perspicacité leur efficacité et leur adaptabilité dans la résolution de problèmes dans divers domaines, notamment celui de la nutrition que nous avons scrupuleusement étudié en amont. En outre, nous avons grandement apprécié la simplicité avec laquelle nous avons pu gérer le problème de la détermination du gagnant en tirant profit des différentes bibliothèques mises à disposition par le framework pour la gestion des agents.