



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et de l'Informatique
Département de Mathématiques et d'Informatique
Filière : Informatique

RAPPORT DE MINI-PROJET DE MASTER EN INFORMATIQUE
Option : Ingénierie des Systèmes d'Information

THEME

Gestion des contraintes SLA à base d'un algorithme bio-inspiré dans l'environnement des Clouds

Etudiant(e)s : M^{elle} « MOUMENE HASNIA »

M^{elle} « ZITOUNI NESRINE »

Encadrant(e) : « Dr. MEROUFEL BAKHTA »

Membre de jury :

Président : Mr « MOUSSA Mouhamed »

Examineur : Mme « SIDI YAKHLEF Soraya »

Année Universitaire 2018/201

Remerciement

Au terme de ce travail, nous aimerons en premier lieu remercier Dieu Allah le tout puissant qui nous a donné la volonté et le courage pour réaliser ce modeste travail.

Nous tenons aussi à exprimer notre profonde gratitude à notre encadreur **Mme MEROUFEL BAKHTA**, nous la remercions de nous avoir encadré, orienté, aidé et conseillé.

Nous adressons nos remerciements aux membres des jurys pour avoir bien voulu examiner et juger ce travail.

Nous tenons également à remercier Monsieur **ZITOUNI CHIHAB EDDINE** pour son aide et pour tout le temps qu'il nous a consacré.

Nos remerciements les plus sincères à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce travail durant tout le long de notre cursus universitaire, non enseignants plus spécialement **Mme FILLALI FATIMA EL ZOHRA**.

Enfin, nous adressons nos remerciements les plus sincères à nos familles, en particulier nos chers parents, pour leurs soutien moral, leurs aide et prières, nos amis, qui nous ont toujours encouragé au cours de la réalisation de ce mémoire.

Merci à toutes et à tous.

Résumé

Le Cloud Computing est Aujourd'hui le sujet phare dans le Domain des Systèmes d'information et de communication, Il fournit des ressources informatiques, et gagne rapidement en popularité entant qu'alternative à l'infrastructure informatique traditionnelle. Alors que de plus en plus de consommateur délèguent leur tâches à des fournisseurs de Cloud. Les accords sur les niveaux de services(SLA) entre client et fournisseurs apparaissant comme un aspect essentiel. On peut dire que la relation entre client et fournisseurs est basé sur les contrats SLA.

Notre travaille étudié les méthodes utilisée pour la création de SLA basée sur la négociation intelligente. Nous proposons aussi une stratégie de négociation intelligente qui hybride entre deux stratégies d'optimisation : PSO et Tabou.

A la fin de ce manuscrit nous avons évalué et les résultats expérimentaux prouvent l'efficacité de l'approche proposée.

Mots clés: Cloud Computing, SLA (accords niveau service), Négociation, Agents, QOS (Qualité de service), PSO, Tabou.

Abstract

Cloud Computing is today the flagship subject in the field of information and communication systems, it provides data processing resources, and is rapidly gaining popularity as an alternative to the traditional data processing infrastructure. As more and more consumers delegate their tasks to cloud providers. Service Level Agreements (SLAs) between customers and suppliers appear to be essential. We can say that the relationship between customer and suppliers is based on SLA.

Our work investigated the methods used for creating SLA based on intelligent trading. We also propose a smart trading strategy that combines two optimization strategies: PSO and Taboo.

At the end of this manuscript we evaluated and the experimental results prove the effectiveness of the proposed approach.

Keywords: Cloud Computing, SLA (Service Level Agreement), Negotiation, Agent, QOS (Quality of Service), PSO, Tabou.

Sommaire

I.	Introduction générale.....	1
II.	Chapitre 1 : le Cloud Computing et SLA.....	2
1.	Introduction.....	2
2.	Définition.....	2
3.	Historique du Cloud Computing.....	2
4.	Services de Cloud Computing.....	3
4.1.	IaaS.....	3
4.2.	PaaS.....	4
4.3.	SaaS.....	4
4.4.	DaaS.....	5
5.	Types de Cloud Computing.....	5
5.1.	Le Cloud Public.....	5
5.2.	Le Cloud Privé.....	6
5.3.	Le Cloud Communautaire.....	6
5.4.	Le Cloud Hybride.....	6
6.	Les avantages et les inconvénients.....	7
6.1.	Les avantages.....	7
6.2.	Les inconvénients.....	8
7.	Définition de SLA.....	8
8.	Historique SLA.....	9
9.	But de SLA.....	10
10.	Caractéristiques de SLA.....	10
11.	Contenus de SLA.....	10
12.	SLA et qualité de service pour le Cloud Computing.....	11
13.	Conclusion.....	11
III.	Chapitre 2 : La négociation.....	12
1	Introduction.....	12
2	SLA et la négociation.....	12
3	Définition de la négociation.....	13
4	SMA et Négociation.....	13
5	Processus de négociation.....	13
6	Types de négociation.....	14
6.1	Négociation Distributive.....	14

6.1.1	Définition.....	14
6.1.2	Bénéfices	14
6.2	Négociation Intégrative	14
6.2.1	Définition.....	15
6.2.2	Bénéfices	15
6.3	Négociation Raisonnée.....	15
6.3.1	Définition.....	15
6.3.2	Principe de base.....	15
7	Méthodes dans la négociation.....	15
7.1	Théorie des jeux	15
7.2	Heuristiques.....	15
7.3	L'argumentation	16
8	Mécanisme de négociation	16
1)	pré négociation	16
2)	négociation	16
9	Avantages de négociation.....	16
10	Inconvénients de négociation	17
11	Conclusion	17
IV.	Chapitre3 : Conception et développement.....	18
1	Introduction	18
2	Méthodes bio-inspirés	18
2.1	Algorithme génétiques.....	18
2.2	Réseaux neurones [32]	19
2.3	La technique d'optimisation par colonie (ACO)	19
3	Contribution	20
4	Optimisation par essaim de particule (PSO).....	20
5	Mode de fonctionnement de l'algorithme PSO	21
5.1	Les éléments de la P.S.O	22
5.2	La notion du voisinage	22
5.3	L'ALGORITHME PSO	24
6	Application de la méthode NEGOp.....	25
6.1	Synthèse et analyse.....	27
7	Description d'algorithme NEGOTPSO	28
7.1	Définition de recherche de Tabou et son rôle dans NEGOTPSO	28

7.2	Description d'algorithme NEGOTPSO.....	28
7.2.1	Pré-Négociation.....	28
7.2.2	Négociation à base de PSO et Tabou.....	31
8	Conclusion	35
V.	Chapitre4 : implémentation et résultats.....	36
1	Introduction	36
2	Environnements de développement.....	36
2.1	Environnement matériel	36
2.2	Environnement Logiciel.....	36
2.2.1	JAVA.....	36
2.2.2	NETBEANS	36
2.2.3	JDK.....	37
2.2.4	JADE	37
3	Les différentes interfaces graphiques	37
4	L'analyse des résultats	44
5	Conclusion	45
VI.	Conclusion générale.....	46

Liste des tableaux

Tableau 1: symbole d'lgorithme de PSO	24
Tableau 2 : symbole de l'algorithme NEGOp.....	26
Tableau 3: Mapping entre PS et NESGO _{TPSO}	29
Tableau 4: symbole de l'algorithme NEG _{TPSO} ,.....	32
Tableau 5: symbole de l'algorithme NEG _{TPSO} ,.....	35

Table des figures

<u>Figure 1 : Types de Cloud Computing</u>	3
<u>Figure 2 : le modèle IaaS</u>	4
<u>Figure 3 : Le modèle PaaS</u>	4
<u>Figure 4 : le modèle SaaS</u>	5
<u>Figure 5: Les différents types d'environnements de déploiement des services de Cloud [6]</u>	6
<u>Figure 6 : Cycle de vie des SLA</u>	9
<u>Figure 7 : Processus de gestion de niveau de service</u>	12
<u>Figure 8 : Le processus de négociation</u>	13
<u>Figure 9 : Types de négociation</u>	14
<u>Figure 10: Volée d'Anser en formation en [38]</u>	20
<u>Figure 11 : Déplacement d'une particule</u>	21
<u>Figure 12 : Un voisinage géographique [42]</u>	23
<u>Figure 13: Un voisinage en cercle [42]</u>	23
<u>Figure 14 Diagramme de classe de négociation</u>	30
<u>Figure 15 diagramme use case de négociation</u>	30
<u>Figure 16 Diagramme de séquence de négociation</u>	31
<u>Figure 17 calculer la distance de fournisseur</u>	33
<u>Figure 18 : calcule la distance dans l'état de client</u>	33
<u>Figure 19 Diagramme d'activité de négociation</u>	34
<u>Figure 20 : Page de démarrage</u>	37
<u>Figure 21: Page d'accueil</u>	38
<u>Figure 22: Fenêtre d'achat</u>	38
<u>Figure 23: Sélection des produits</u>	39
<u>Figure 24: Message d'erreurs</u>	39
<u>Figure 25: Page de négociation de client</u>	40
<u>Figure 26 : Paramètre donnée par le client</u>	40
<u>Figure 27: Page de négociation de fournisseur</u>	41
<u>Figure 28: Paramètre données par le fournisseur</u>	41
<u>Figure 29: Page des résultats finals</u>	42
<u>Figure 30: Résultat de la liste de tabou de client et de fournisseur</u>	42
<u>Figure 31: Résultats obtenus d'un exemple</u>	43
<u>Figure 32: Affichage des graphes des résultats de chaque itérations</u>	43
<u>Figure 33 : Résultat entre NEGOp et NEGOpso</u>	44
<u>Figure 34 : Nombre de succès avec la population</u>	44
<u>Figure 35 : degré de succès</u>	45

I. Introduction générale

La technologie de l'Internet se développe de manière exponentielle depuis sa création.

Actuellement, une nouvelle tendance a fait son apparition dans le monde des technologies de l'information et de la communication, il s'agit du Cloud Computing. Cette technologie, s'appuyant sur le WEB 2.0, offre des occasions aux sociétés de réduire les coûts d'exploitation des logiciels par leurs utilisations directement en ligne.

Le terme Cloud Computing, ou « informatique dans les nuages », est un nouveau modèle informatique qui consiste à proposer les services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui. Cette nouvelle technologie permet à des entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosse quantité d'informations.

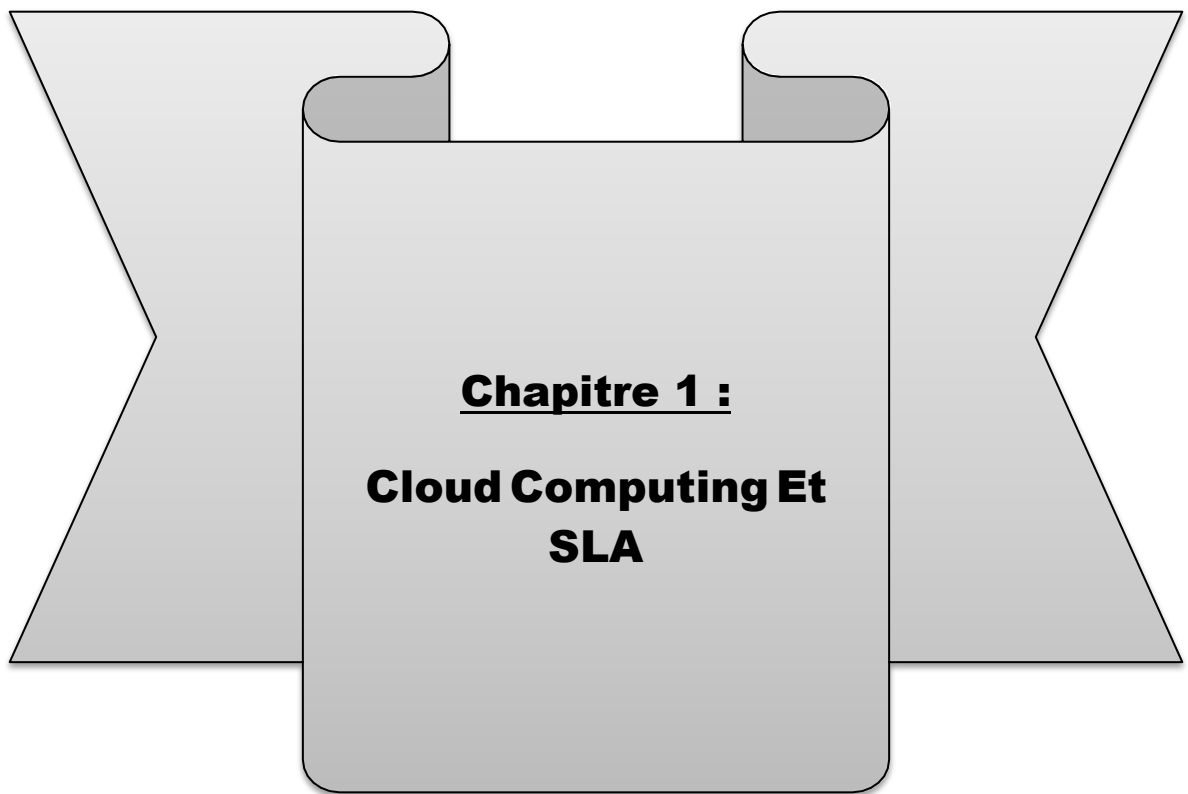
Le Cloud Computing basé sur des contrats de niveau de service (SLA) qui s'étendent à travers ces nuages du fait de l'importance de l'évaluation de la qualité du service et en raison de la multiplication de services similaires au Cloud. Ce contrat contribue à établir une relation de confiance et de partenariat entre le client et le fournisseur. Il permet de simplifier les problèmes complexes, ainsi que fournir un cadre général de compréhension des deux parties (Clients et Fournisseurs).

Considérant que le SLA est un contrat négocié, le processus de négociation permet d'améliorer les accords sur des points de vue communs ou des plans d'actions grâce à l'échange structuré d'information pertinente.

Dans le cadre de la réalisation de notre projet, on a choisi le Cloud Computing comme un environnement de travail pour étudier des protocoles de négociation d'accord de niveau de service basé sur des méthodes et des mécanismes d'optimisation, visant à obtenir un équilibre et résoudre des conflits entre client et fournisseur.

Ce document est organisé en quatre chapitres :

- Le premier chapitre va porter les différentes notions de base de Cloud computing avec la précision de diverses informations de SLA.
- Le deuxième chapitre Explique les principes et les différents concepts de négociation.
- Le troisième chapitre présente notre approche de négociation NEG_{PSOT} à base de PSO et Tabou. Ce chapitre explique en détaille les différentes étapes et traitements en utilisant la modélisation par le langage UML.
- Enfin un dernier chapitre " D'implémentation " », aborde la mise en œuvre de notre application et qui aboutira sur une conclusion générale.



Chapitre 1 :

**Cloud Computing Et
SLA**

1. Introduction

Les technologies de l'information et de la communication évoluent et révolutionnent nos modes de vie et de travail. Le Cloud computing ou informatique virtuelle, est apparu ces dernières années comme un nouveau modèle de gestion et d'utilisation des systèmes informatiques. Le concept consiste à déporter sur des serveurs distants les traitements et stockages habituellement effectués en local afin d'y accéder sous forme de service.

Dans ce premier chapitre nous allons donner des idées générales sur le fonctionnement et les fonctionnalités du Cloud Computing avec la présentation des différentes notions de SLA (service level agreement) en précisant ses liens avec le Cloud.

2. Définition

Le Cloud Computing est un terme général employé pour désigner la livraison de ressources et de services à la demande par internet. Il représente un paradigme informatique dans lequel un grand nombre de systèmes sont connectés à des réseaux privés ou publics, afin de fournir une infrastructure évolutive de manière dynamique pour le stockage d'application, de données et de fichiers.

Il est également possible de transformer un centre de données d'un environnement à forte intensité de capital en un environnement à prix variables.

L'idée est basée sur un principe très fondamental de « réutilisation de capacités informatique » et que le Cloud n'est pas un ensemble de technologie, mais un modèle de fourniture, de gestion et de consommation des services et des ressources informatiques localisés dans Datacenter. La différence apportée par le Cloud Computing par rapport aux concepts traditionnels de « calcul en grille », « calcul distribué », « calcul utilitaire » ou « calcul autonome » consiste à élargir les horizons au-delà des frontières organisationnelles [1].

3. Historique du Cloud Computing

Il est communément admis que le concept de Cloud Computing a été initié par le géant Amazon en 2002. Le cybermarchand avait alors investi dans un parc informatique afin de pallier les surcharges des serveurs dédiés au commerce en ligne constatées durant les fêtes de fin d'année. A ce moment-là, Internet comptait moins de 600 millions d'utilisateurs mais la fréquentation de la toile et les achats en ligne étaient en pleine augmentation. En dépit de cette augmentation, les ressources informatiques d'Amazon restaient peu utilisées une fois que les fêtes de fin d'année étaient passées. Ce dernier a alors eu l'idée de louer ses capacités informatiques le reste de l'année à des clients pour qu'ils stockent les données et qu'ils utilisent les serveurs. Ces services étaient accessibles via Internet et avec une adaptation en temps réel de la capacité de traitement, le tout facturé à la consommation.

Cependant, ce n'est qu'en 2006 qu'Amazon comprit qu'un nouveau mode de consommation de l'informatique et d'internet faisait son apparition. Bien avant la naissance du terme de Cloud Computing, utilisé par les informaticiens pour qualifier l'immense nébuleuse du net, des services de Cloud étaient déjà utilisés comme le webmail2, le stockage de données en ligne (photos, vidéos,...) ou encore le partage d'informations sur les réseaux sociaux.

La virtualisation est un concept beaucoup plus ancien qui constitue le socle du Cloud Computing.

La virtualisation regroupe l'ensemble des techniques matérielles ou logicielles permettant de faire fonctionner, sur une seule machine physique, plusieurs configurations informatiques (systèmes d'exploitation, applications, mémoire vive,...) de manière à former plusieurs machines virtuelles qui reproduisent le comportement des machines physiques [2].

4. Services de Cloud Computing

On peut distinguer quatre principaux services de Cloud Computing: IaaS pour le matériel, PaaS pour le middleware, SaaS pour les logiciels et les applications et DaaS

(Voir Figure 1) [3].

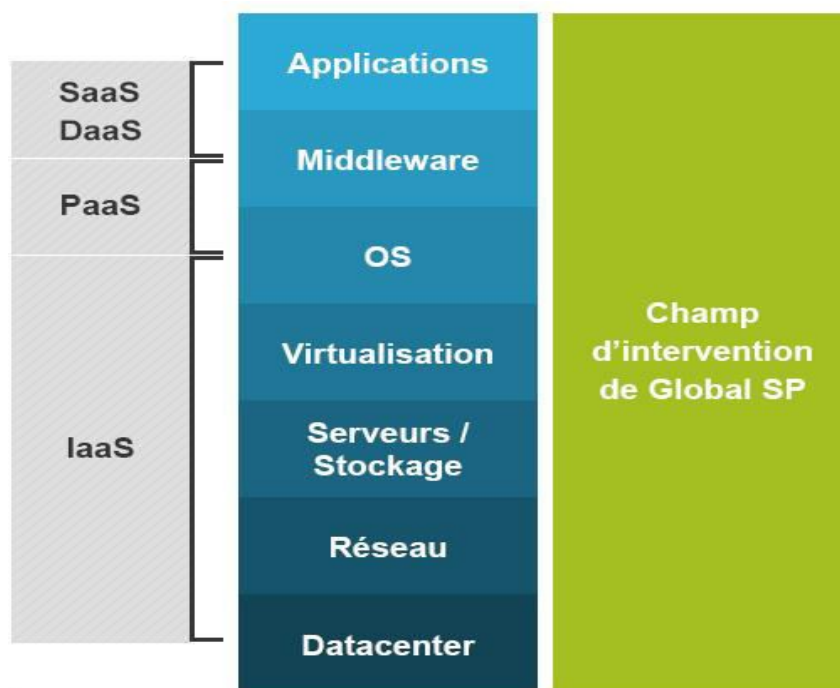


Figure 1 : Types de Cloud Computing.

4.1. IaaS

L'IaaS (Infrastructure As A Service) désigne la couche basse du Cloud Computing et peut être considéré comme un point d'entrée dans les technologies du Cloud.

Dans ce modèle, seul le matériel (serveurs, baies de stockage, réseau, etc.), socle d'infrastructure est hébergé chez un prestataire ou un fournisseur, comme Amazon AWS, Microsoft Azure etc. L'entreprise qui fait ce choix bénéficie d'une infrastructure partagée et automatisée. L'avantage, par rapport à une infrastructure classique hébergée chez elle est de payer seulement pour ce que l'on consomme et bénéficier de toutes fonctionnalités offertes par le prestataire, comme une garantie de SLAs, la sécurité, etc [3].

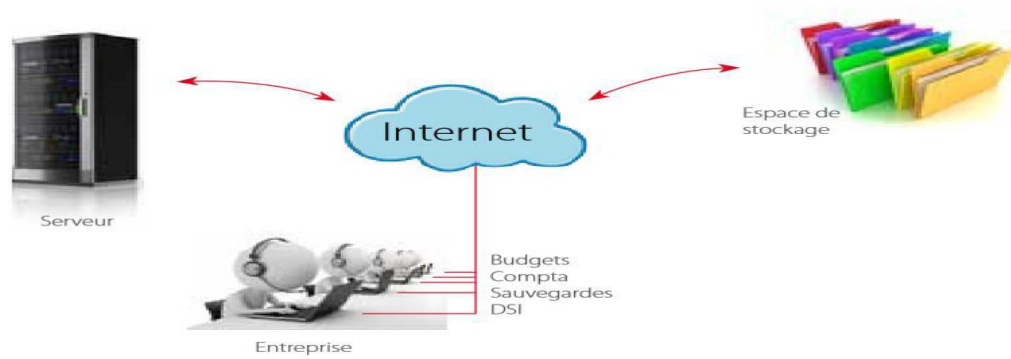


Figure 2 : le modèle IaaS.

Le schéma illustré dans la figure 2 [4] représente le contenu d'IaaS dont pour des calculs complexes ou ponctuels (simulations budgétaires, calculs prédictifs...) ou encore du stockage (notamment dans le cadre de sauvegardes), les administrateurs de ressources informatiques trouvent dans le Cloud des environnements aux capacités quasi illimitées et dont la mise en œuvre est quasi instantanée.

4.2. PaaS

Le modèle PaaS (Plateforme As A Service) se place sur le niveau supérieur de l'IaaS en offrant aux entreprises un environnement leur permettant de déployer leurs développements. Le PaaS fournit ainsi des langages de programmation, des bases de données et différents services pour faire fonctionner leurs applications. De plus, il automatise entièrement le déploiement (mises à jour, correctifs, etc.) et la montée en charge. En général, les entreprises qui ont besoin pour leurs développements de PaaS le combinent souvent avec du IaaS. Ils bénéficient ainsi d'une plateforme de développement entièrement sécurisée et automatisée et accessible à la demande. Ils ne paient ainsi que ce qui est consommé [3].



Figure 3 : Le modèle PaaS.

La figure 3 [4] représente la gestion de contrôle dans le modèle PaaS. Donc L'utilisateur gère, mais ne contrôle pas l'infrastructure Cloud (réseaux, serveurs, systèmes d'exploitation, stockage). Il a ainsi le contrôle sur les applications déployées et la possibilité de configurer l'environnement d'hébergement applicatif.

4.3. SaaS

Le modèle SaaS (Software As A Service) fournit des applications à l'utilisateur sous la forme d'un service prêt à l'emploi. C'est en fait un nouveau modèle de consommation des applications. Pour les entreprises, le SaaS peut être totalement découplé des infrastructures et

donc du IaaS et du PaaS. D'ailleurs, beaucoup de sociétés font leurs premiers pas dans le nuage par le biais des applications SaaS : les outils de messagerie ou de bureautique comme Gmail ou office365 sont de bons exemples de SaaS ! Il est donc amusant de constater que la plupart des employés d'une entreprise utilisent souvent des applications SaaS sans le savoir. Pour l'entreprise, l'avantage est certain : les applications SaaS ne nécessitent aucune maintenance, les mises à jour étant régulièrement faites par l'éditeur, et le paiement est à la demande [3].

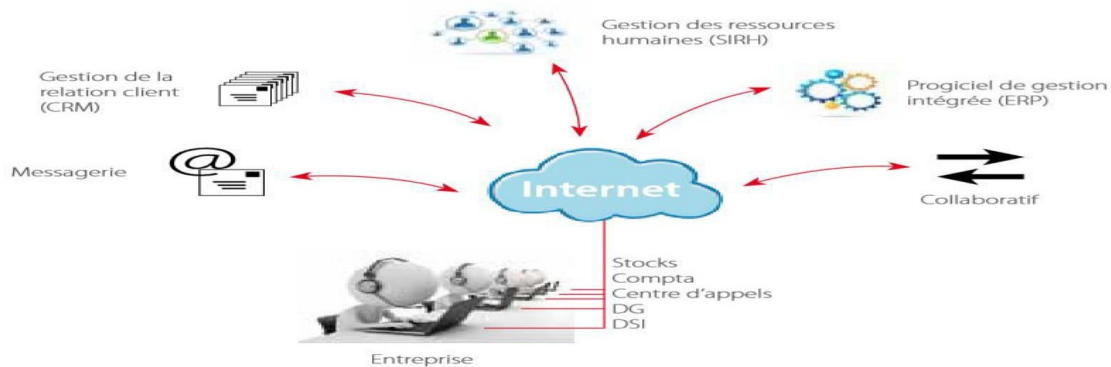


Figure 4 : le modèle SaaS.

La figure 4 [4] représente les différentes gestions de SaaS avec les autres modèles de Cloud.

De nombreux logiciels sont disponibles dans le Cloud, en mode SaaS :

- Messagerie : L'e-mail est certainement l'application la plus utilisée en mode Cloud
- CRM : La gestion de la relation client est l'autre application phare du Cloud
- ERP : Certains progiciels de gestion intégrée sont proposés en mode SaaS.
- Collaboratif : Les outils de collaboration (partage de documents, réseaux sociaux... se prêtent bien au mode SaaS.

4.4. DaaS

DaaS (Desktop as a Service) aussi appelé en français bureau en tant que service, bureau virtuel est une technologie de virtualisation qui permet aux entreprises d'accéder à un bureau virtuel via un portail web sécurisé. L'ensemble des applications métiers et des ressources de l'utilisateur est hébergé dans le Cloud et accessible en un clic avec une simple connexion à Internet. Ce service Cloud fonctionne sur la base d'une souscription par abonnement [3].

5. Types de Cloud Computing

On doit commencer par déterminer le type de déploiement Cloud ou d'architecture de Cloud Computing sur lequel vos services Cloud seront implémentés. Il existe trois modes de déploiement de services Cloud : le Cloud public, le Cloud privé et le Cloud hybride.

5.1. Le Cloud Public

Un IaaS opéré par un fournisseur de nuage dit « public », partagé et accessible à la demande.

Cloud « mutualisé » ouvert à tous, type ceux d’Amazon, Microsoft, etc. Cloud étant une infrastructure de serveurs virtualités hautement disponible et multi-Datacenter [5].

5.2. Le Cloud Privé

L’infrastructure Cloud fonctionne pour une organisation unique. Elle peut être gérée par l’organisation elle-même (Cloud Privé interne) ou par un tiers (Cloud Privé externe). Dans ce dernier cas, l’infrastructure est entièrement dédiée à l’entreprise et accessible via des réseaux sécurisés de type VPN [5].

Cloud monté pour une utilisation dédiée à un unique client. Les Clouds privés internes, gérés en interne par une entreprise pour ses besoins, les Clouds privés externes, dédiés aux besoins propres d’une seule entreprise, mais dont la gestion est externalisée chez un prestataire [5].

5.3. Le Cloud Communautaire

L’infrastructure est partagée par plusieurs organisations qui ont des intérêts communs (par exemple les exigences de sécurité, de conformité ...). Comme le Cloud Privé, il peut être géré par les organisations elles-mêmes ou par un tiers.

Un Cloud communautaire est utilisé par plusieurs organisations qui ont des besoins communs. Ainsi cela peut porter sur l’hébergement d’une application métier très spécialisée, mais commune à de très nombreuses entreprises, qui décident de fédérer leurs efforts [5].

5.4. Le Cloud Hybride

L’infrastructure se compose de deux nuages ou plus (Privé, Communautaire ou Public), qui restent des entités uniques, mais qui sont liées par une technologie normalisée ou propriétaire. Ceci permet la portabilité des données ou des applications [5].

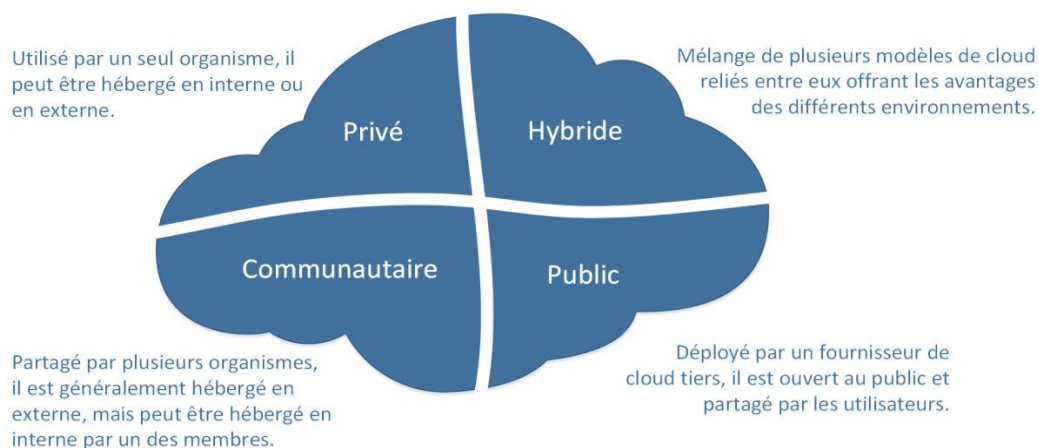


Figure 5: Les différents types d’environnements de déploiement des services de Cloud [6]

6. Les avantages et les inconvénients

Le Cloud a joué un rôle important l'avenir de la technologie de l'entreprise et du stockage des données. Il offre la possibilité de mettre à l'abri leurs données à des prix abordables aux entreprises et aux particuliers. Mais malgré ces privilèges, le Cloud présente bien des inconvénients, qui constituent un véritable handicap pour les utilisateurs.

6.1. Les avantages

Les avantages de Cloud computing sont comme suit :

- **La réduction des coûts**

La mutualisation des ressources informatiques et la facturation à l'usage rend le Cloud Computing économiquement attrayant.

- **L'accessibilité**

Les services de Cloud Computing sont accessible à tout moment, sur tous les supports, via une connexion internet.

- **L'élasticité**

Le Cloud Computing permet d'allouer simplement et rapidement davantage de ressources à des applications en production afin de répondre à des montées en charge ponctuelles.

- **Le déploiement rapide et la simplicité d'intégration**

Le déploiement et la mise en fonctionnement d'un service de Cloud Computing nécessite peu de temps.

- **La disponibilité du service**

Le Cloud Computing permet de garantir les accès et la disponibilité des services. Le fournisseur de services de Cloud Computing s'engage contractuellement sur une interruption minimum des serveurs à travers des SLA (service LevelAgreements).

- **La flexibilité nécessaire pour vos projets**

Le Cloud Computing s'adapte en temps réel à vos projets et accompagne le développement d'une activité sans coûts supplémentaires.

- **L'adoption rapide par les utilisateurs finaux**

Les applications utilisant des services de Cloud Computing sont pour la plupart faciles à adopter. Le Cloud Computing simplifie les usages.

- **La conséquence positive sur l'environnement**

Le Cloud Computing, basé sur la virtualisation de serveur, la mutualisation de la puissance de calcul et la flexibilité des services s'inscrit dans une démarche éco-responsable.

- **La réversibilité**

La restitution de l'intégralité des données d'une entreprise est garantie par les fournisseurs prévoyant dans leur contrat une clause de réversibilité [7].

6.2. Les inconvénients

Le Cloud Computing a aussi des inconvénients :

- **La bande passante**

Peut faire exploser le budget La bande passante qui serait nécessaire pour mettre cela dans le Cloud est gigantesque, et les coûts seraient tellement importants qu'il est plus avantageux d'acheter le stockage nous-mêmes plutôt que de payer quelqu'un d'autre pour s'en charger.

- **Les performances des applications**

Peuvent être amoindries Un Cloud public n'améliorera définitivement pas les performances des applications.

- **La fiabilité du Cloud**

Un grand risque lorsqu'on met une application qui donne des avantages compétitifs ou qui contient des informations clients dans le Cloud.

- **Taille de l'entreprise**

Si l'entreprise est grande alors ses ressources sont grandes, ce qui inclut une grande consommation du Cloud. Du coup il serait peut-être plus intéressant à mettre au point son propre Cloud plutôt que d'en utiliser un externalisé. Les gains sont bien plus importants quand on passe d'une petite consommation de ressources à une consommation plus importante.

- **La sécurité du Cloud**

La sécurité vis-à-vis du stockage : si les données sont conservées dans un seul disque, ou si elles sont entre plusieurs unités de stockage.

La sécurité et la confidentialité des données : si le fournisseur de service assure des tests portant sur sa sécurité informatique et si de tes tests sont faits de façon régulières.

La sécurité des locaux : sont-ils inaccessibles pour des personnes malintentionnées ?

- **Le piratage**

Certaines applications comme Facebook et Twitter sont très sujets aux attaques. Le piratage d'un compte d'entreprise pourrait avoir des conséquences néfastes pour la réputation de l'entreprise, tandis que l'utilisation imprudente des applications par un salarié pourrait offrir aux cybercriminels l'opportunité d'entrer dans le réseau et de soustraire des données des clients.

- **L'optimisation des applications**

Malgré une connexion internet rapide, avec un débit garanti, certaines applications web peuvent s'avérer être très lentes. Elles peuvent s'avérer être plus limitées que des applications fonctionnant sur les propres ordinateurs de l'entreprise [8].

7. Définition de SLA

Le terme "SLA" est l'abréviation de "Service Level Agreement". En français, cette expression peut être traduite par « accord de niveau de service », « convention de services », ou encore « engagements de service ».

Le SLA est un accord négocié, et un contrat qui quantifie le niveau de service minimal pour une prestation qu'un fournisseur s'engage à délivrer à son client. Il peut être soit une partie d'un contrat informatique, soit une annexe à un contrat informatique, soit une annexe à des conditions générales [9].

8. Historique SLA

L'utilisation de l'entente de service est apparue pendant les années 1980 pour gérer la qualité de service dans le domaine des télécommunications. En effet, grâce aux services utilitaires comme le grid Computing (grille de services), le SLA et en particulier ses services commerciaux sous-jacents ont fortement émergé.

Plus récemment, les départements informatiques de certaines grandes entreprises ont repris l'idée et utilisent les SLA avec leurs clients — généralement, des utilisateurs d'autres départements au sein de l'entreprise — pour permettre une comparaison entre la qualité de service fournie et celle promise et le cas échéant substituer le prestataire par un autre [9].

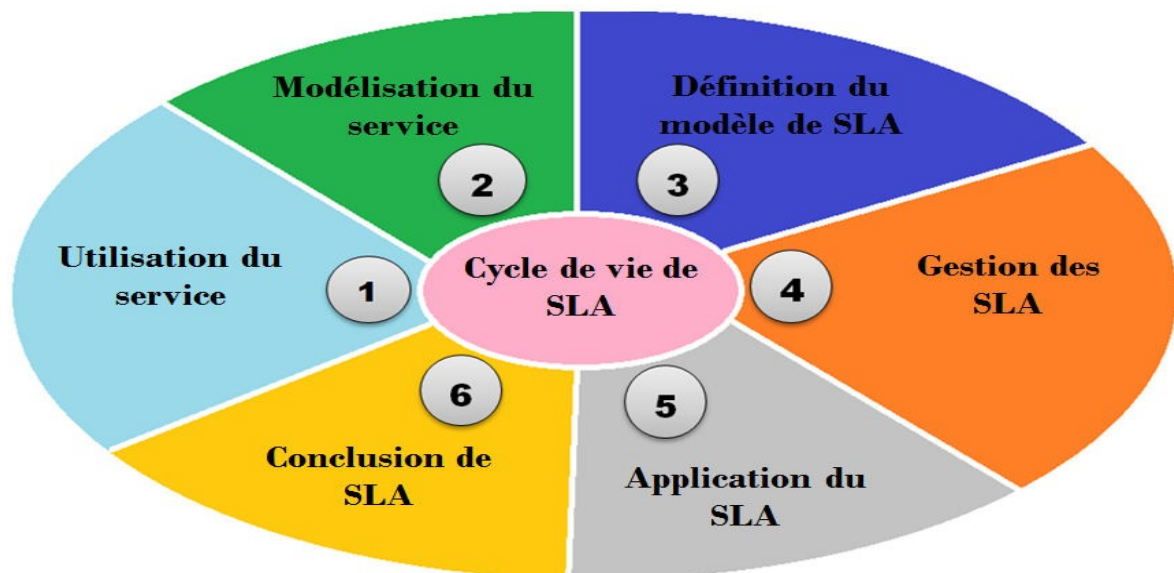


Figure 6 : Cycle de vie des SLA

Le cycle de vie des contrats de niveau de service, illustré à la figure 6 [10], comprend six phases principales: utilisation du service(1), modélisation du service(2), définition du modèle de contrat de niveau de service(3), gestion des contrats de niveau de service(4), application du contrat de niveau de service(5) et conclusion du contrat de niveau de service(6). Les phases sont numérotées de 1 (première phase) à 6 (dernière phase) sur la figure 1.

- La première phase traite des informations qui affectent l'utilisation du service Cloud par le consommateur de service.
- La deuxième phase traite de la modélisation du service, des relations et des dépendances au sein des composants du service et des informations relatives à la fourniture du service.
- Dans la troisième phase, des modèles de contrat de niveau de service sont créés et d'autres informations connexes sont capturées.

- La quatrième phase traite de la gestion des accords de niveau de service couvrant divers aspects tels que la définition des accords de niveau de service, la modélisation des accords de niveau de service, la négociation des accords de niveau de service (y compris la renégociation des accords de service après la fourniture de services dans le nuage), la surveillance des accords de niveau de service, l'évolution et la violation des contrats.
- Le but de la cinquième phase est de faire respecter le SLA.
- La sixième phase traite de la résiliation du contrat SLA, ce qui peut se produire pour diverses raisons, telles que la violation du contrat SLA et / ou l'expiration de la période de service.

9. But de SLA

Les principaux buts d'un SLA sont de définir :

- les besoins d'un client pour pouvoir les exprimer clairement d'une manière compréhensible par chacune des parties (fournisseur et client).
- les critères d'évaluation ainsi que les moyens de mesure avec lesquels on pourra évaluer la qualité de la prestation fournie.

Les autres buts sont entre autres :

- d'établir une relation de confiance, voire de partenariat entre les parties.
- d'éliminer les attentes irréalistes ou trop chères (rapport prix/prestation) [11].

10. Caractéristiques de SLA

Les SLA mesurent les performances et la qualité du fournisseur de services de différentes manières. Ainsi, un SLA peut spécifier les éléments de mesure ou indicateurs suivants

- Disponibilité Pourcentage de temps durant lequel les services sont disponibles.
- Nombre d'utilisateurs pouvant être pris en charge simultanément.
- Bancs d'essai de performances spécifiques à l'aune desquels sont mesurées périodiquement les performances réelles.
- Temps de réponse des applications.
- Calendrier des notifications préalables à des modifications du réseau susceptibles d'affecter les utilisateurs.
- Délai de réponse du service d'assistance pour différentes catégories de problèmes [9].

11. Contenus de SLA

La rédaction de cet engagement de service change selon le type de service proposé. Toutefois, son contenu inclut généralement les éléments cités ci-dessous :

- Les dates de début et de fin du contrat.
- La description, la liste et l'étendue des services.
- Le temps imparti et calendrier des interventions.
- Le temps de réponse du prestataire de service en cas de panne ou d'accroc.

- Les utilisateurs du service.
- Le prix de la prestation ainsi que les conditions de paiement.
- Le taux de disponibilité du serveur.
- Le droit applicable et la propriété intellectuelle.
- La nécessité de prodiguer des informations et des conseils du professionnel.
- La sécurité et la récupération des données [11].

12.SLA et qualité de service pour le Cloud Computing

Le Cloud Computing est un modèle qui permet l'accès à la demande et à distance à un ensemble de ressources P calcul configurables. Ce modèle tend à fournir des services logiciels et matériels à l'utilisateur tout en minimisant l'effort en termes d'installation, configuration et de maintenance de ces services, à la fois pour l'utilisateur et pour le fournisseur de Cloud.

La qualité de service et du contrat de niveau de service (Service Level Agreement – SLA) dans le Cloud soulève des problèmes de performance, de disponibilité, de consommation énergétique et de coûts économiques du Cloud. Nous pensons qu'un des éléments différenciateurs entre les offres de Cloud sera la qualité du service fournie et le SLA proposé par le Cloud.

Le point de vue de l'utilisateur de Cloud et le point de vue du fournisseur de Cloud sont les deux piliers du projet, et leur complémentarité une des originalités du projet. Ainsi, deux objectifs principaux seront poursuivis. D'une part, d'un point de vue du fournisseur de Cloud, comment construire un Cloud qui soit capable de se reconfigurer dynamiquement et de manière autonome pour garantir le niveau de service et le SLA établi avec l'utilisateur.

D'autre part, d'un point de vue de l'utilisateur de Cloud, comment fournir à l'utilisateur un moyen de gouvernance de SLA pour qu'il soit informé de toute violation de SLA, ainsi que de l'empreinte énergétique du Cloud au cours du temps. Une approche novatrice, combinant la théorie de la commande avec les langages et algorithmes distribués, sera proposée pour la reconfiguration autonome de Cloud et la gouvernance de son SLA [12].

13.Conclusion

Le Cloud Computing est l'accès via un réseau de télécommunications, à la demande et en libre-service, à des ressources informatiques partagées configurables. Il s'agit donc d'une délocalisation de l'infrastructure informatique. Les SLA s'étendent à travers les nuages d'informations du fait de l'importance de l'évaluation de la qualité du service et en raison de la multiplication de services similaires au Cloud Computing. Le SLA doit spécifier un ou plusieurs objectifs de niveau de services précédemment définis afin de garantir que la QoS (ou qualité de service) délivrée a satisfait les attentes du consommateur.

Dans le chapitre suivant nous allons montrer des techniques de négociation utilisées pour bien gérer la partie de discussion entre les fournisseurs et les clients.



1 Introduction

Le Cloud Computing doit disposer de normes bien définies en ce qui concerne la qualité de service(QoS), la gestion et la surveillance des accords de niveau de service. Dans la partie précédente nous avons définis le contrat de niveau de service (SLA) dans le Cloud. La négociation est un outil pour atteindre un contrat qui satisfait les besoins de QoS des deux Acteurs de Cloud Computing : Fournisseurs et Clients. Nous nous intéressons dans ce chapitre à la définition et les mécanismes de négociation.

2 SLA et la négociation

Les accords de niveau de service (SLA) sont spécifiés dans la bibliothèque ITIL (Information Technology Infrastructure Library) dans le cadre d'un concept plus vaste appelé Service Level Management (SLM) [8]. Ce processus est un processus en cinq étapes: négociation, finalisation, surveillance, rapport, révision. Dans ce processus, un contrat de niveau de service est la base utilisée pendant toute la fourniture du service, comme illustré à la figure 7.

Les exigences de niveau de service (SLR) sont définies précédemment et constituent le point de départ de SLM. Ici, les reflex sont conçus par une partie et sont utilisés comme base pour la phase de négociation. En conséquence, un accord sur les niveaux de service est défini à l'étape de finalisation: un service est instancié après accord des deux parties. Ce service doit être observé dans la phase de surveillance où les mesures sont fournies et sont utilisées pour l'évaluation d'un accord. Ici, les mesures sont nécessaires pour montrer que le service fourni est conforme ou non au SLA de l'étape du rapport. Le processus se termine après l'étape de révision, suivie d'un processus d'amélioration du service (SIP).

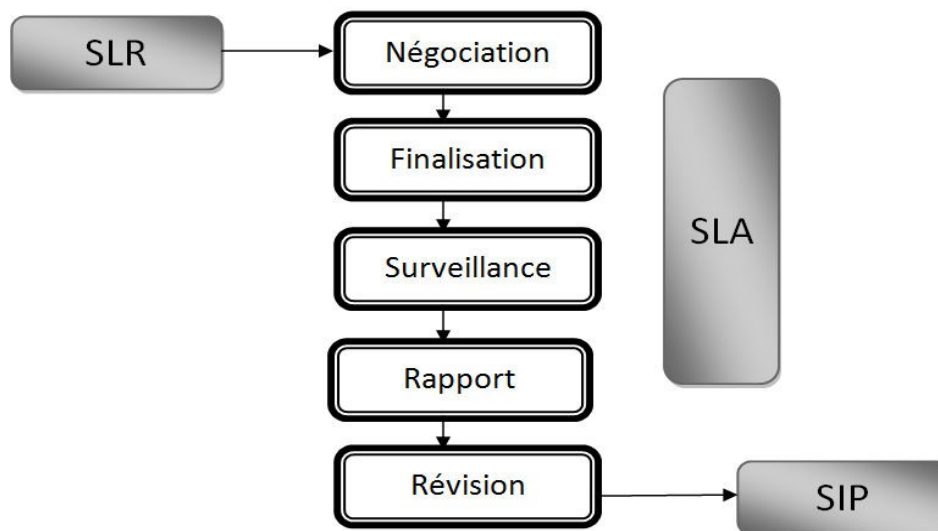


Figure 7 : Processus de gestion de niveau de service.

3 Définition de la négociation

Selon le dictionnaire de Petit Larousse : « La négociation est l'action de négocier, de discuter les affaires communes entre des parties en vue d'un accord » [9].

Selon l'Équipe Perspective monde : « La négociation est un processus de communication et d'échanges entre au moins deux parties dont l'objet concerne l'organisation d'une relation ou le règlement d'une problématique entre celles-ci» [10].

La négociation est un processus par laque une décision commune à deux agents au plus est prise; chacun d'entre eux essayant d'atteindre leurs buts au objectifs personnels [11].

4 SMA et Négociation

La négociation dans un système multi-agent correspond à un processus de recherche d'accord entre plusieurs agents. L'accord peut concerner un prix, une ressource ou tout simplement un point de rencontre. Le processus de recherche peut faire intervenir:

- a) des échanges d'informations,
- b) la restriction des buts initiaux des agents,
- c) des concessions mutuelles.

Le terme «négociation» est utilisé ici dans le sens de recherche de consensus. Quand on parle de négociation entre agents, on doit d'abord définir le point de départ à partir duquel, les agents vont poli voir négocier. Ce point est appelé rencontre. Chaque agent est alors défini par l'ensemble des buts qu'il veut atteindre. L'entente est le consensus sur lequel les agents se seront mis d'accord après négociation [12].

5 Processus de négociation

Le processus de négociation permet de trouver un arrangement et un profit mutuel pour tous les agents négociateurs. La Figure 8 décrit le processus de négociation [13].

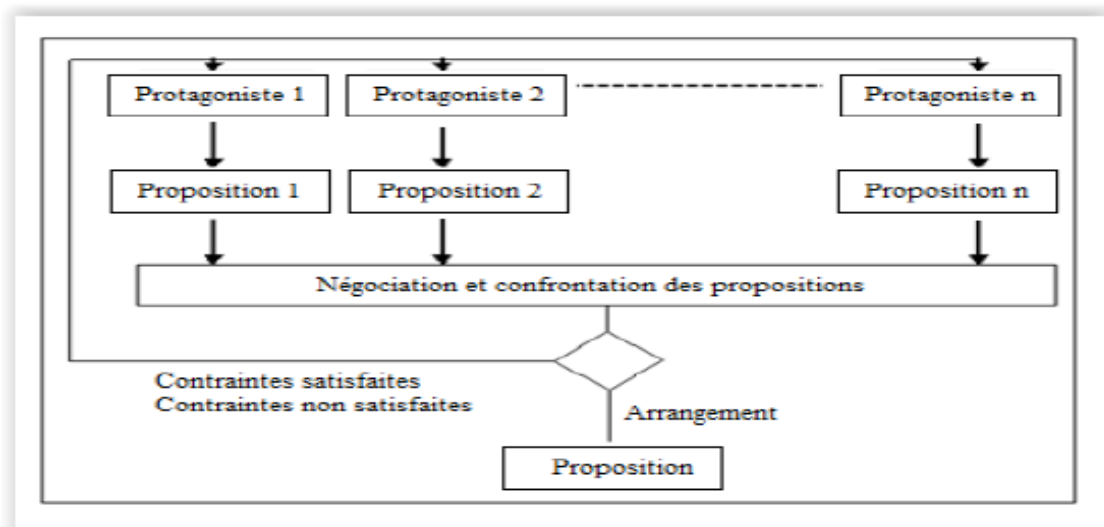


Figure 8 :Le processus de négociation.

Dans la Figure 8 chaque protagoniste envoie sa proposition à l'initiateur concernant l'exécution d'une même tâche donnée. Le processus de négociation commence dès la réception de toutes les propositions. Si l'une de ces propositions répond aux objectifs de toutes les parties négociatrices, il y aura un arrangement et la négociation sera terminée. Dans le cas contraire, les contraintes satisfaites et non satisfaites seront envoyées aux négociateurs afin de reformuler de nouvelles propositions. Le processus de négociation se répète jusqu'à ce qu'un arrangement mutuel acceptable soit atteint. En absence d'un arrangement, l'initiateur arrête la négociation par un échec.

6 Types de négociation

Il existe trois différents types de négociation qui représente dans la figure 9 [14]:

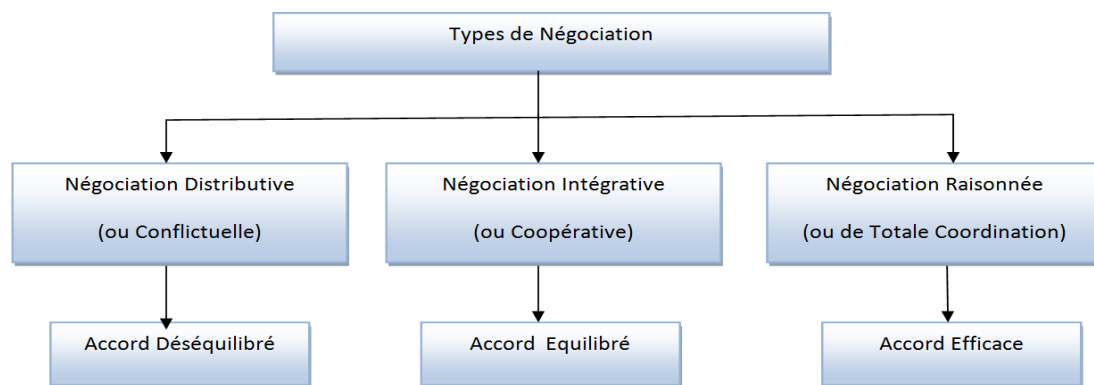


Figure 9 : Types de négociation.

6.1 Négociation Distributive

Négociation Distributive ou conflictuelle a une logique conflictuelle: gagnant-perdant.

6.1.1 Définition

Négociation dans laquelle les négociateurs font preuve d'une coopération fiable (ou à la limite, inexistante). Ils sont animés par un désir de gain propre, réalisé au détriment des objectifs adverses ou communs [15].

6.1.2 Bénéfices

- Possibilité de tirer profit du rapport de force.
- Favorise les négociateurs habiles et expérimentés.
- Simplifier les considérations stratégiques (intérêts personnels seulement).
- Maximisation des gains à court terme [15].

6.2 Négociation Intégrative

La négociation coopérative ou intégrative a une logique gagnant-gagnant.

6.2.1 Définition

Négociation dans laquelle les négociateurs font preuve d'une forte coopération et sont animés par un désir de gain mutuel [16].

6.2.2 Bénéfices

- Engendre un climat de confiance, crédibilité et de réciprocité.
- Encourage une plus grande stabilité de l'accord.
- Favorise des relations à long terme.
- Valorise la créativité et les actions constructives.
- Aide à l'enrichissement de la culture d'entreprise [16].

6.3 Négociation Raisonnée

Négociation Raisonnée ou de totales coordinations a une logique coopérative/efficace : gagnant-gagnant.

6.3.1 Définition

La négociation raisonnée est un mode de résolution des conflits. Les négociateurs se concentrent sur des intérêts communs afin de trouver un accord rationnel sur la base de critères objectifs [17].

6.3.2 Principe de base

Il y a quatre principes de base:

- Traiter de façon séparée les questions de personnes.
- Se concentrer sur les intérêts en jeu et non sur les positions.
- Explorer plusieurs pistes de solution.
- Rechercher une entente basée sur des critères objectifs et vérifiable [17].

7 Méthodes dans la négociation

Chaque mécanisme décisionnel de négociation automatisée peut adopter une approche parmi les trois suivantes: la négociation basée sur la théorie des jeux, la négociation basée sur les heuristiques et la négociation basée sur l'argumentation.

7.1 Théorie des jeux

La théorie des jeux représente un outil pour l'étude des stratégies d'interaction entre des agents autonomes dans une négociation automatisée. Plusieurs travaux ont utilisé la théorie des jeux pour la conception des processus de négociations automatisés dans différents domaines [18] [19]. Ces travaux tendent à déterminer la stratégie optimale à utiliser pour l'atteinte d'un équilibre d'une simulation d'un jeu entre des agents négociateurs identiques. Cette approche permet aux agents d'adopter un comportement rationnel lorsqu'ils prennent leurs décisions et leurs choix de stratégies. Les agents adoptent toujours dans la théorie des jeux un comportement rationnel. Ce type de comportement peut parfois les amener à des solutions non satisfaisantes.

7.2 Heuristiques

Ces agents optent pour des stratégies approximatives et heuristiques [20][21]. Les heuristiques sont des règles qui ne fournissent pas nécessairement la solution optimale, mais elles permettent de fournir une solution plus proche de l'optimal. L'approche heuristique est basée sur le test et l'évaluation des différents résultats. Néanmoins, l'approche basée sur les heuristiques présente un certain nombre de limites. En effet, elle ne conduit pas nécessairement à une solution optimale, car elle utilise des stratégies approximatives et elle n'examine pas tout l'espace possible des résultats. Cette approche ne peut pas prévoir le comportement exact du système et ceux de ses agents et elle suppose que l'agent possède une connaissance complète de ses désirs et de ses préférences.

7.3 L'argumentation

Dans un contexte de la négociation basée sur l'argumentation, l'argument est définie comme l'information qui permet à l'agent: (1) de justifier sa position dans la négociation et (2) d'influencer les positions des autres agents [22]. La justification permet aux agents de donner des informations supplémentaires dans leurs propositions. Ils peuvent expliquer les raisons du rejet ou de l'acceptation d'une proposition donnée. Cependant, l'agent négociateur à travers cette approche ne peut pas intercepter et identifier toutes les différentes sortes de conflits existants dans les arguments.

8 Mécanisme de négociation

Un processus de négociation est habituellement composé de deux grandes étapes :

- 1) pré négociation (ou planification de la négociation) qui se rapporte aux discussions qui précèdent les négociations formelles et qui inclut souvent des questions de procédure : qui sera impliqué, où et quand les négociations auront lieu, comment elles seront structurées, quel sera l'objet de la négociation ?
Les réponses aux certaines de ces questions seront des valeurs pour les dimensions définies auparavant, comme le participant ou le temps. Afin de répondre aux autres questions des nouvelles dimensions et les possibles valeurs doivent être définies.
- 2) négociation : qui se rapporte aux interactions avec l'échange des propositions/contrepropositions formulées à partir des stratégies de négociation.

9 Avantages de négociation

- La négociation est probablement le mode de règlement des conflits le plus flexible puisque n'y participent que les intéressés, de même que leur représentant, le cas échéant. Les parties sont libres de préparer les négociations en conformité avec leurs propres besoins.
- La négociation n'offre aucune garantie. Toutefois, selon plusieurs commentateurs, les négociations sont plus susceptibles de réussir lorsque les parties adoptent une approche fondée sur les intérêts par opposition à une approche fondée sur les positions.
- La négociation est un processus volontaire. Personne n'est forcé d'y participer.
- Il n'est pas nécessaire d'avoir recours à une tierce partie impartiale. Il est important de le rappeler, car il arrive que les parties préfèrent qu'aucune personne étrangère ne participe au processus.

- En prenant pour acquis que les parties négocient de bonne foi, la négociation leur permettra de préparer un accord qui tient compte de leurs intérêts.
- Les négociations peuvent permettre de préserver et, dans certains cas, d'améliorer les rapports entre les parties lorsqu'une entente a été conclue entre elles.
- La négociation pourrait s'avérer moins coûteuse et moins longue qu'un procès [23].

10 Inconvénients de négociation

- Une négociation particulière peut entraîner un résultat positif. Toutefois, les parties n'ont peut-être pas le même pouvoir et la partie la plus faible pourrait se trouver dans une moins bonne position.
- Une négociation qui réussit exige que chacune des parties comprenne clairement le mandat qui lui a été confié. Si l'une des parties comprend mal les limites de son pouvoir en matière de négociation, elle ne pourra pas participer de façon efficace au processus.
- L'absence d'une tierce partie impartiale peut amener une partie à tenter d'abuser l'autre.
- Aucune partie ne peut être contrainte de poursuivre les négociations. Toute personne qui souhaite mettre fin aux négociations peut le faire à tout moment au cours du processus nonobstant le temps et l'effort qui ont pu être investis par l'autre partie.
- Le processus de négociation ne peut garantir la bonne foi d'une partie ni le fait qu'elle soit digne de confiance.
- La négociation peut servir de stratégie pour gagner du temps et empêcher une autre partie de faire valoir ses droits (notamment par un procès ou l'arbitrage) [24].

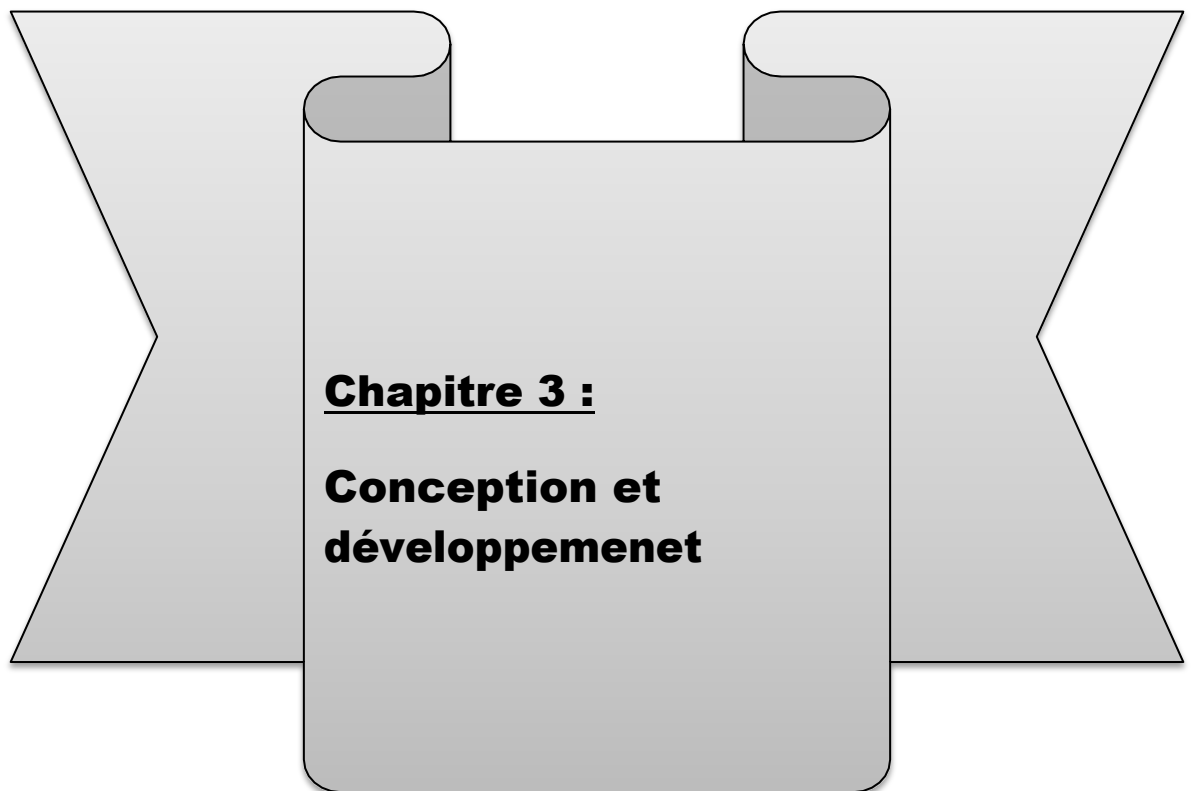
11 Conclusion

Une négociation intervient lorsque des agents interagissent pour prendre des décisions communes, alors qu'ils poursuivent des buts différents plus précisément, pour l'objectif de la négociation est de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs.

Dans le cadre de ce chapitre nous avons présentés les différents points de négociation et les techniques utilisées pour permettre résoudre ses conflits.

La négociation dans l'environnement de cloud computing est généralement contrôlée par le temps et les ressources, ce qui rend le choix de la stratégie de négociation critique pour sélectionner les meilleurs paramètres SLA.

Dans le chapitre suivant, nous nous présentons un travail sur la négociation basée sur un algorithme bio inespéré et nous proposeront une amélioration de cet algorithme afin de le rendre plus efficace et utile.



1 Introduction

L'homme s'inspire de plus en plus de la nature qui l'entoure pour mettre en place des algorithmes simulant le comportement des animaux. Les Méta-heuristiques permettent de trouver facilement et rapidement la solution la plus approchée de l'optimum si ce dernier existe. Nous allons présenter dans ce chapitre quelques méthodes concernant ce domaine et étudier la stratégie « optimisation par essaims de particule » présentant ses limites et on termine avec notre proposition d'une approche pour résoudre les problèmes de négociation intelligente.

2 Méthodes bio-inspirés

Les approches bio inspirées sont les plus anciennes et les plus populaires, créées pour deux objectifs principaux:

- Le premier consiste à la modélisation des systèmes naturels et leur simulation sur ordinateur.
- Le deuxième implique l'étude des phénomènes naturels pour développer des systèmes informatiques et des algorithmes aptes à résoudre des problèmes complexes [29].

Les algorithmes d'optimisations inspirés de la nature peuvent être déterministes ou stochastiques (probabiliste). Les méthodes formelles qui ont été utilisées pour résoudre des problèmes d'optimisations exigent d'énormes efforts de calcul, qui tendent à échouer avec l'augmentation de la taille des problèmes [30].

On peut citer parmi les approches les plus connues : les algorithmes génétiques, les réseaux neurones, les techniques d'optimisation par colonies de Fourmies (ACO) et L'optimisation par essaims particulaires (PSO).

2.1 Algorithme génétiques

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes, ils sont des algorithmes d'optimisations stochastiques fondées sur les mécanismes de la sélection naturelle et de la génétique. Ils ont été initialement développés par John Holland en 1975 [30].

Un algorithme génétique implémente une version très simplifiée et très schématique des mécanismes de l'évolution biologique.

Il est défini par quatre éléments de bases suivantes [31] :

- ✓ Individu/chromosome/séquence : une solution potentielle du problème qui correspond à une valeur codée de la variable (ou des variables) en considération.
- ✓ Population : un ensemble de chromosomes ou de points de l'espace de recherche (donc des valeurs codées des variables).
- ✓ Environnement : l'espace de recherche (caractérisé en termes de performance correspondant à chaque individu possible).
- ✓ Fonction de performance (fitness) : la fonction - positive - que nous cherchons à maximiser, car elle représente l'adaptation de l'individu à son environnement.

L'avantage d'algorithme génétique est leur simplicité. Il représente plusieurs solutions potentielles, ce qu'il permet d'explorer plusieurs zones dans l'espace de recherche.

Néanmoins, ils ne sont pas très efficaces dans la résolution d'un problème, ils sont coûteux en temps de calcul. Ces algorithmes n'offrent aucune garantie.

2.2 Réseaux neurones [32]

Réseau neurone est considéré comme un système de traitement de l'information conçu avec une inspiration qui provient du système nerveux. Son but est de développer des machines capables de mémoriser des expériences puis de les analyser pour prendre des décisions à propos de situations nouvelles.

Parmi toutes les techniques bios inspirées, le réseau neurone constitue le domaine le plus ancien et le plus vaste. Il permet, comme leur principe indique, de calculer toute fonction calculable.

Ils se caractérisent par la représentation distribuée et le traitement parallèle de l'information (connaissance). Et du point de vue conceptuel ils peuvent être caractérisés par :

- Un ensemble de neurones artificiels, appelés également nœuds, unités, ou tout simplement neurones.
- La forme de la connectivité entre les neurones, appelée l'architecture ou la structure du réseau.
- Une méthode pour déterminer les valeurs des poids, appelée leurs algorithmes d'entraînements ou d'apprentissage.

Les RN sont des outils statiques, leur grand avantage réside dans leur capacité d'apprentissage automatique, ce qui permet de résoudre des problèmes sans avoir recours à l'écriture des règles complexes tout en étant tolérant aux erreurs.

Mais ils sont des boîtes noires où les connaissances sont inintelligibles pour l'utilisateur, et en cas d'erreurs système, il est quasiment impossible d'en déterminer la cause.

2.3 La technique d'optimisation par colonie (ACO) :

La technique d'optimisation par colonies de fourmis a été introduite par Marc Dorigo dans sa thèse au début des années 1990[33].

Elle représente un algorithme d'optimisation inspiré de la nature, qui est motivé par le comportement naturel de recherche de nourriture des espèces de fourmis [30].

Dans l'ACO, chaque fourmi est un agent qui construit une solution et la piste de phéromones représente une information qui sera utilisée de façon probabiliste pour faire évoluer les fourmis-solutions. Dans le contexte d'ACO, on considère que les fourmis se déplacent sur un graphe valeur [34].

L'algorithme de colonies de fourmis offre beaucoup de souplesse, il est possible de l'adapter à tous les grands problèmes combinatoires classiques. En plus, il se parallélise de façon très naturelle [30].

Malgré ses avantages, il comporte certaines limites. Alors, il ne fonctionne pas bien quand un grand nombre d'arcs sur le graphe de construction font partie des bons chemins qui ont des valeurs de probabilité égale.

3 Contribution :

Le but de notre travail est de trouver une solution Bio-inspiré qui permet de résoudre le problème de conflit en respectant le temps de négociation avec un résultat mutuellement satisfaisant.

Nous avons choisi « $NEGO_{TPSO}$ » (Négociation A base de Tabou & PSO) comme un nom de notre proposition qui représente une amélioration de la méthode proposée dans l'article [25] et qui nous avons la nommée « $NEGO_P$ ».

La méthode $NEGO_{TPSO}$ corrige les erreurs et résoudre les problèmes existants dans $NEGO_P$ en utilisant le recherche Tabou avec la technique d'essais et aussi la stratégie de compromis (trade-off).

D'abord, nous montrons la stratégie $NEGO_P$ présentée dans l'article [25] avec ses différents principes et limites. Ensuite, nous présentons notre approche proposé « $NEGO_{TPSO}$ » qui analyse et traite les problèmes avec l'élimination des défauts trouvés dans $NEGO_P$.

4 Optimisation par essaim de particule (PSO) :

L'Optimisation par Essaim de Particules (OEP) ou ParticleSwarmOptimization (PSO) en anglais, est un algorithme inscrit dans la famille des algorithmes évolutionnaires. Il a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio psychologue) en 1995 [35].

Cette méthode trouve sa source dans les observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons de Reynold [37], Heppner & Grenander [36]. Autrement-dit, elle s'inspire fortement de l'observation des relations grégaires d'oiseaux migrateurs, qui pour parcourir de « longues distances » (migration, quête de nourriture, parades aériennes, etc.), doivent optimiser leurs déplacements en termes d'énergie dépensée, de temps, (etc.), comme par exemple la formation en forme de V présentée dans la Figure 10.



Figure 10: Volée d'Anser en formation en [38]

Le déplacement de ses animaux en essaim est complexe, sa dynamique obéit à des règles et des facteurs bien spécifiques qu'il s'agit de cerner :

- Chaque individu dispose d'une certaine intelligence « limitée » (qui lui permet de prendre une décision).
- Chaque individu doit connaître sa position locale et disposer d'information locale de chaque individu se trouvant dans son voisinage
- Obéir à ces trois règles simples, « rester proche des autres individus », « aller dans une même direction » ou « voler à la même vitesse ».

Tous ses facteurs et règles sont indispensables pour le maintien de la cohésion dans l'essaim, ceci par l'adoption d'un comportement collectif complexe et adaptatif.

5 Mode de fonctionnement de l'algorithme PSO

La population dans l'algorithme PSO est nommée essaim, chaque individu du groupe est dit particule. Le déplacement de toute particule (comme indiqué ci-dessus) est régi par des règles et conditions bien spécifiques, influencé par le mouvement des autres particules du voisinage.

Dans un tel contexte ce déplacement à une signification et doit parallèlement répondre à une logique, fondement même du PSO. Il consiste à chercher un optimum dans un voisinage donné, ce déplacement est influé par les trois composantes suivantes :

- Une composante d'inertie : la particule s'efforce de suivre instinctivement son cap de déplacement.
- Une composante cognitive : la particule fait tout pour se diriger vers la meilleure position rencontrée jusqu'à présent.
- Une composante sociale : la particule s'inspire également de l'expérience, du parcours des autres particules, pour se diriger vers la meilleure position rencontrée par ses voisins

Ce déplacement est illustré dans la Figure 11.

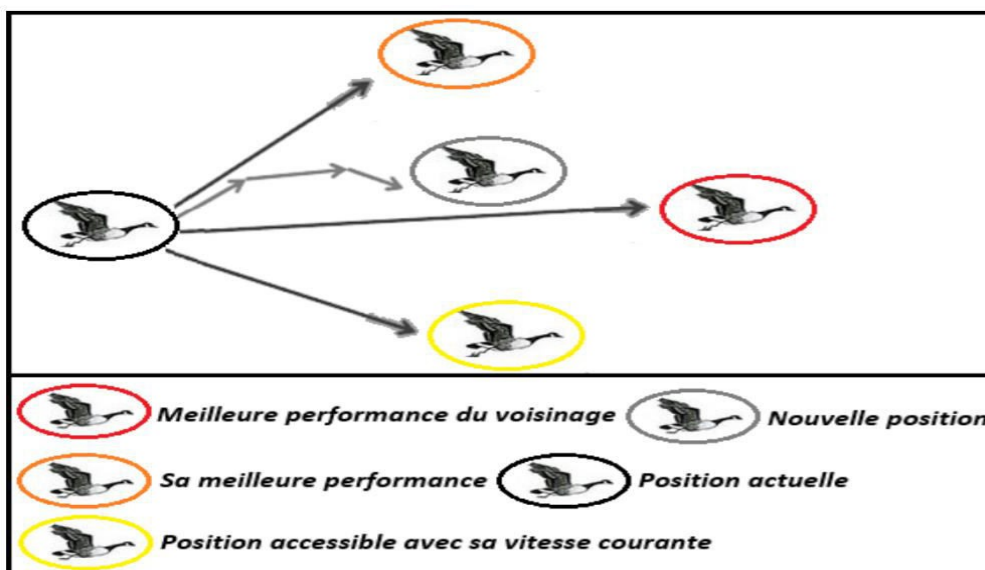


Figure 11 : Déplacement d'une particule

A terme, on se rend compte que toutes les particules, après un certain nombre d'itérations, convergent vers une même position, somme toute la meilleure rencontrée par l'ensemble des particules. Cela ne signifie pas pour autant, dans l'absolu, que c'est la meilleure position du voisinage, juste que c'est la meilleure rencontrée [39].

5.1 Les éléments de la P.S.O

Pour appliquer la PSO il faut définir un espace de recherche constitué de particules et une fonction objectif à optimiser. Le principe de l'algorithme est de déplacer ces particules afin qu'elles trouvent l'optimum.

Chacune de ces particules est dotée :

- D'une position, c'est-à-dire ses coordonnées dans l'ensemble de définition.
- D'une vitesse qui permet à la particule de se déplacer. De cette façon, au cours des itérations, chaque particule change de position. Elle évolue en fonction de son meilleur voisin, de sa meilleure position, et de sa position précédente. C'est cette évolution qui permet de tomber sur une particule optimale.
- D'un voisinage, c'est-à-dire un ensemble de particules qui interagissent directement sur la particule, en particulier celle qui a le meilleur critère.

A tout instant, chaque particule connaît :

- Sa meilleure position visitée. On retient essentiellement la valeur du critère calculée ainsi que ses coordonnées.
- La position du meilleur voisin de l'essaim qui correspond à l'ordonnement optimal.
- La valeur qu'elle donne à la fonction objectif car à chaque itération il faut une comparaison entre la valeur du critère donnée par la particule courante et la valeur optimale [40].

D'après Maurice Clerc et Patrick Siarry [41], l'évolution d'une particule n'est finalement qu'une fusion de trois types de comportements :

- Egoïste (se déplacer suivant sa vitesse actuelle).
- Conservateur (revenir en arrière en prenant en compte sa meilleure performance).
- Panurgien (suivre aveuglement le meilleur de tous en considérant sa performance).

On voit alors que la bio-inspiration à l'origine de l'optimisation par essaim particulaire ressort dans l'algorithme sous la forme d'une intelligence collective : coordination du groupe, instinct individuel et interaction locale entre les individus (grognements, phéromones...).

5.2 La notion du voisinage

Le voisinage d'une particule est le sous-ensemble de particules de l'essaim avec lequel il a une communication directe. Ce réseau de rapports entre toutes les particules est connu comme la sociométrie, ou la topologie de l'essaim.

Il existe deux principaux types de voisinage :

- a) Les voisinages géographiques : les voisins sont considérés comme les particules les plus proches. Cependant, à chaque itération, les nouveaux voisins doivent être recalculés à partir d'une distance prédéfinie dans l'espace de recherche. C'est donc un voisinage dynamique (Voir Figure 12).

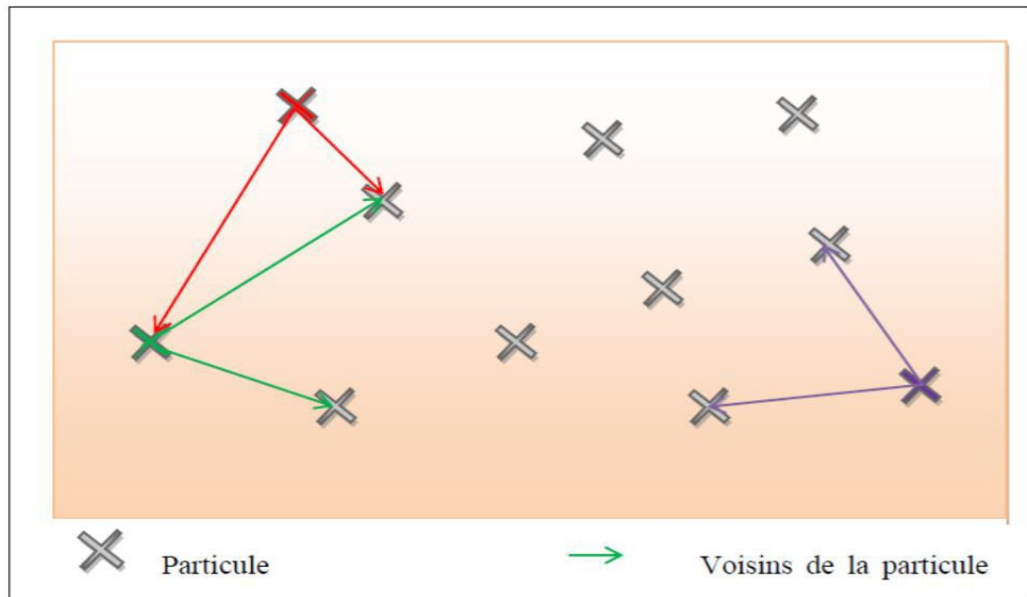


Figure 12 : Un voisinage géographique [42]

Dans cet exemple, le voisinage de la particule est composé des deux particules les plus proches.

- b) Les voisinages sociaux : les voisins sont définis à l'initialisation et ne sont pas modifiés ensuite. C'est le voisinage le plus utilisé, pour plusieurs raisons : Il est plus simple à programmer. Il est moins coûteux en temps de calcul. En cas de convergence, un voisinage social tend à devenir un voisinage géographique. Pour ce faire, on dispose (virtuellement) les particules en cercle puis, pour la particule étudiée, on inclut progressivement dans ses informatrices, d'abord elle-même, puis les plus proches à sa droite et à sa gauche, jusqu'à atteindre la taille voulue.

On peut aussi choisir les informatrices au hasard.

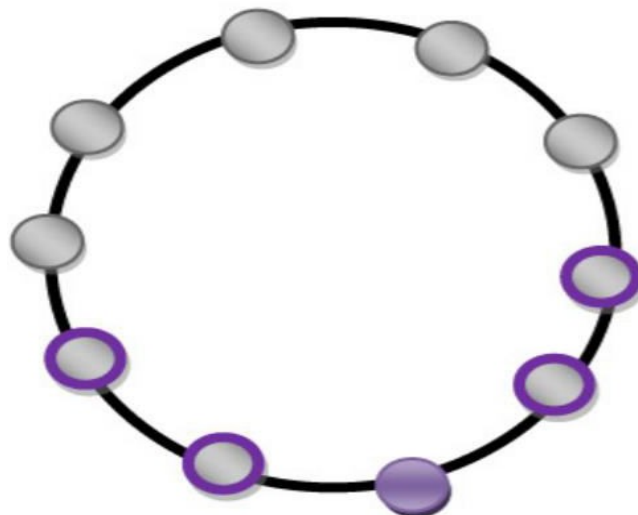


Figure 13: Un voisinage en cercle [42]

Dans l'exemple illustré dans la Figure 13, la particule principale est en bas et ses informatrices correspondent au deux particules directement à sa droite et à sa gauche [40].

5.3 L'ALGORITHME PSO

L'algorithme à base de la P.S.O. travaille sur une population appelée essaim de solutions possibles, elles-mêmes appelées particules. Ces particules sont placées aléatoirement dans l'espace de recherche de la fonction objectif.

Une particule i de l'essaim dans un espace de dimension D est caractérisée, à l'instant t par les paramètres cités dans la le tableau 1.

X	sa position dans l'espace de recherche.
V	sa vitesse.
Pb	la position de la meilleure solution par laquelle elle est passée.
Pg	la position de la meilleure solution connue de tout l'essaim.
f(Pb)	la valeur de fitness de sa meilleure solution.
F(Pg)	la valeur de fitness de la meilleure solution connue de tout l'essaim.

Tableau 1: symbole d'lgorithme de PSO

Le déplacement de la particule i entre les itérations t et $t+1$ se fait selon les deux équations (1) et (2) :

$$V(t+1) = V(t) + C1r1 (Pb(t) - X(t)) + C2r2 (Pg(t) - X(t)) \tag{1}$$

$$X(t+1) = X(t) + V(t+1) \tag{2}$$

Tel que :

- C1 et C2 : deux constantes qui représentent les coefficients d'accélération, elles peuvent être non constantes dans certains cas selon le problème d'optimisation posé.
- r1 et r2 : deux nombres aléatoires tirés de l'intervalle [0,1].

L'algorithme de base de la méthode PSO (Algorithme1) commence par une initialisation aléatoire des particules dans leur espace de recherche, en leurs attribuant une position et une vitesse initiales. À chaque itération de l'algorithme les particules se déplacent selon les équations (1) et (2) et les fonctions objectif (fitness) des particules sont calculées afin de pouvoir calculer la meilleure position de toutes Pg. La mise à jour des Pb et Pg est faite à chaque itération suivant l'algorithme. Le processus est répété jusqu'à satisfaction du critère d'arrêt.

Algoritme1 : Algorithme de PSO
<p>Début</p> <p>Initialiser les paramètres et la taille S de l'essaim ;</p> <p>Initialiser les vitesses et les positions aléatoires des particules dans chaque dimension de l'espace de recherche ;</p> <p>Pour chaque particule, Pb = X ;</p>

```

Calculer f(X) de chaque particule ;
Calculer Pg ; // la meilleure Pb
Tant que (la condition d'arrêt n'est pas vérifiée) faire
    Pour (i allant de 1 à S) faire
        Calculer la nouvelle vitesse à l'aide de l'équation (1) ;
        Trouver la nouvelle position à l'aide de l'équation (2) ;
        Calculer f(X) de chaque particule ;
        Si f(X) est meilleure que f(Pb) alors
            Pb = X ;
        Si f(Pb) est meilleure que f(Pg) alors
            Pg = Pb ;
    Fin pour
Fin tant que
Fin

```

Il existe plusieurs paramètres qui interviennent et influencent la performance de la méthode PSO. Le choix de ces paramètres reste critique et dépend généralement du problème posé, mais a une grande influence sur la convergence de l'algorithme. Parmi les paramètres qui rentrent en ligne de compte citons :

- La dimension du problème.
- Le nombre des particules.
- La disposition des particules.
- Les coefficients de confiance.
- La vitesse maximale.
- Le facteur d'inertie.
- Le facteur de construction.
- La notion du voisinage.
- Le critère d'arrêt.

6 Application de la méthode NEGOp

Les auteurs de travail [25] offrent un mécanisme de négociation à base de PSO (NEGOp). Cette dernière représente un modèle bilatérale, multi-issu, sans médiateur, temps-dépendant.

NEGOp est utilisée par les deux parties de négociation, à savoir le fournisseur et le client, pour calculer les offres à échanger au cours du processus de négociation. Chaque agent (Client/Fournisseur) remplit son essaim de particules proches de l'objectif défini.

Dans un premier temps, le client envoie son objectif sous forme d'offre et, après avoir déterminé les souhaits du fournisseur, il s'ensuit une période d'atténuation au cours de laquelle chaque agent essaie d'imposer ses propres exigences. Après cela, les deux agents passent par une période de coopération au cours de laquelle ils transigent et parviennent à un accord.

Pour bien expliquer la stratégie de NEGOp, on a modélisé un algorithme de négociation avec des symboles choisis (Comme représenté au tableau).

$O_{i,x}$	Offre i d'agent x
E_x	Essaim d'agent x
U_x	Fonction d'utilité d'agent x
SD_x	Seuil dynamique d'agent x
S_{min_x}	Seuil statique d'agent x
$R_x(k)$	Ensemble des offres affectés par la contre-offre envoyée à l'agent x dans la k eme itération
T	Temps actuel
D	Deadline de négociation
$CO_x(k)$	Contre-offre proposé par x pendant la k eme itération

Tableau 2 : symbole de l'algorithme NEGOp

Tout d'abord, un opérateur de sélection est défini afin de déterminer les individus de l'essaim modifiés par l'offre de l'agent opposé. Une offre de l'agent opposé agit comme un vent pour une particule dans l'essaim et modifie sa vitesse et sa distance en fonction d'un facteur de compromis. Le nombre des offres concernées par la contre-offre de l'agent opposé varie avec le temps: moins de particules (offres) sont sélectionnées au début du processus de négociation et leur nombre augmente avec le temps. De cette manière, l'agent est plus têtù au début du processus de négociation, car moins de particules dans l'essaim sont affectées par l'offre de l'agent opposé qui devient de plus en plus importante avec le temps (voir algorithme2).

Algorithme2 : L'algorithme de NEGOp
λ facteur d'accélération.
<u>Construction d'essaims :</u>
While $i \leq N$ do \ N: Nombre des particules
if $U_x(O_{i,x}) \geq S_{min_x}$ then
$O_{i,x} \in E_x$

```

End if
End while

$$CO_x(k) = \frac{\sum_{i \in EX}^n O_{i,x}}{\#EX}$$

    Envoie COx(k) (k) à contre agent y
A la réception de COx(k) par y :
If Ux (COi, x) ≥ S Dy then
    Accepte COi, x
Else
    
$$A = \frac{\#S_y \times (D-t)}{D}$$

End if
While i ≤ A do
    Ry (k) [i] = Sy[i]
End while
While i < #Ry (k) do \ changeement de vitesses
    Vi, t+1 = λ (COx (k) – Ry(k)[i])
    Ry(k) [i] = Ry(k) [i] + Vi,t+1
End while
    
```

6.1 Synthèse et analyse

La méthode de négociation NEGOp [25] a réussi de résoudre les conflits et gérer la négociation entre le client et le fournisseur en respectant le contrainte de temps. Mais cette stratégie a des problèmes causés par la méthode PSO et même par la négociation elle-même. Parmi ces problèmes :

1. Problème d’optimal local : C’est le cas où une offre sera bloquée dans une valeur fixe causée par le voisinage.
2. Problème de calcul lourd : le calcul des coefficients de PSO peut augmenter la surcharge de négociation
3. Problème de voisinage non déterminé : le travail [25] ne détermine pas combien de voisin ou comment déterminer les voisins pour chaque offre.

Afin de résoudre ces problèmes ; nous proposerons dans la section suivante notre stratégie de négociation NEGOTPSO .

7 Description d'algorithme NEGOTPSO

Notre approche de négociation nommée NEGOTPSO (NEGOCIation à base de Tabou et PSO) est une version améliorée de NEGOP. Dans NEGOTPSO, nous avons essayé de résoudre les problèmes cités dans la section précédentes par les manières suivantes :

- Pour résoudre le problème d'optimal local, nous avons utilisé la recherche de Tabou pour éviter le blocage et les boucles infinies des itérations de négociations avec les mêmes offres.
- Pour résoudre le problème de calcul lourd, nous avons utilisé la stratégie de compromis entre l'offre et ses voisins (cette stratégie était déjà utilisée dans des PSO adaptive mais pas dans la négociation)
- Pour résoudre le problème de voisinage, nous avons utilisé un voisinage dynamique à base de temps de négociation resté et le nombre des offres dans l'essaim.

7.1 Définition de recherche de Tabou et son rôle dans NEGOTPSO

La recherche Tabou est une méta-heuristique itérative qualifiée de «recherche locale» au sens large. L'idée de la recherche Tabou consiste, à partir d'une position donnée, à explorer le voisinage et à choisir le voisin qui minimise la fonction objectif.

Elle est utilisée pour la résolution des problèmes d'optimisation, destinée principalement à guider d'autres méthodes afin de trouver de meilleures solutions à partir d'une solution initiale obtenue par l'une des heuristiques.

Le principe de base de cet algorithme est de pouvoir poursuivre la recherche de solutions même lorsqu'un optimum local est rencontré et ce, en permettant des déplacements qui n'améliorent pas la solution, et en utilisant le principe de la mémoire pour éviter les retours en arrière (mouvements cycliques) [40].

Dans NEGOTPSO, chaque négociateur (Client/Fournisseur) utilise un tableau de Tabou local qui stocke les offres déjà traité ou refusé par l'adversaire. Cette stratégie a deux avantages :

- Eviter l'optimal local
- Eviter de renvoyer une offre déjà refusé par l'adversaire.

7.2 Description d'algorithme NEGOTPSO

L'algorithme de NEGOTPSO est divisé en deux étapes : la pré-négociation et la négociation, chaque étape est expliquée en détail dans les sections suivantes.

Dans notre travail, nous allons présenter les différents diagrammes et modèles de conception en utilisant le langage UML pour modéliser les deux étapes de négociation.

7.2.1 Pré-Négociation :

Chaque agent (client/fournisseur) doit fixer les paramètres de négociation avant de commencer la procédure de négociation elle-même. Parmi ces paramètres :

- ❖ L'intervalle des offres acceptées.
- ❖ Le nombre de population dans l'essaim
- ❖ Les valeurs des offres dans la population

- ❖ Le seuil de valeur de fitness qui désigne les offres acceptées ou pas
- ❖ Le timer.

Tous ces paramètres sont privés pour chaque adversaire, sauf le timer qui sera commun entre eux.

Le tableau 5 montre les correspondances entre un PSO classique et notre approche.

PSO	NEGO _{TPSO}
Essaim	Ensemble des offres
Particule	Offre
Vitesse	Degré de renonciation
Position	Valeur d'offre
Fitness	Qualité d'offre
G _{best}	Meilleur offre dans l'essaim
P _{best}	Meilleur offre local

Tableau 3: Mapping entre PS et NESGO_{TPSO}

La figure 14 représente l'enchaînement général de notre gestion de négociation entre deux agents (Client/fournisseur) en diagramme de classe:

Le diagramme de classe est une représentation statique des éléments qui composent un système et leurs relations dans la modélisation UML. Il est le point central dans une modélisation orientée objet. Il décrit la structure des entités manipulées par l'utilisateur [45].

Dans le diagramme, il y a 6 classes :

- **Personne** : Elle représente une classe des utilisateurs de système (client/fournisseur).
- **Client** : classe héritant de personne contient les informations sur le client.
- **Fournisseur** : classe héritant de personne possède les informations de fournisseur.
- **Offre** : Elle contient les données de l'offre déposée par le client et le fournisseur.
- **Produit** : Cette classe contient les informations trouvées des produits d'achat.
- **NEGO_{TPSO}**: Regroupe les données utilisées par le client et le fournisseur dans le traitement, et fait le calcul du résultat final de négociation.

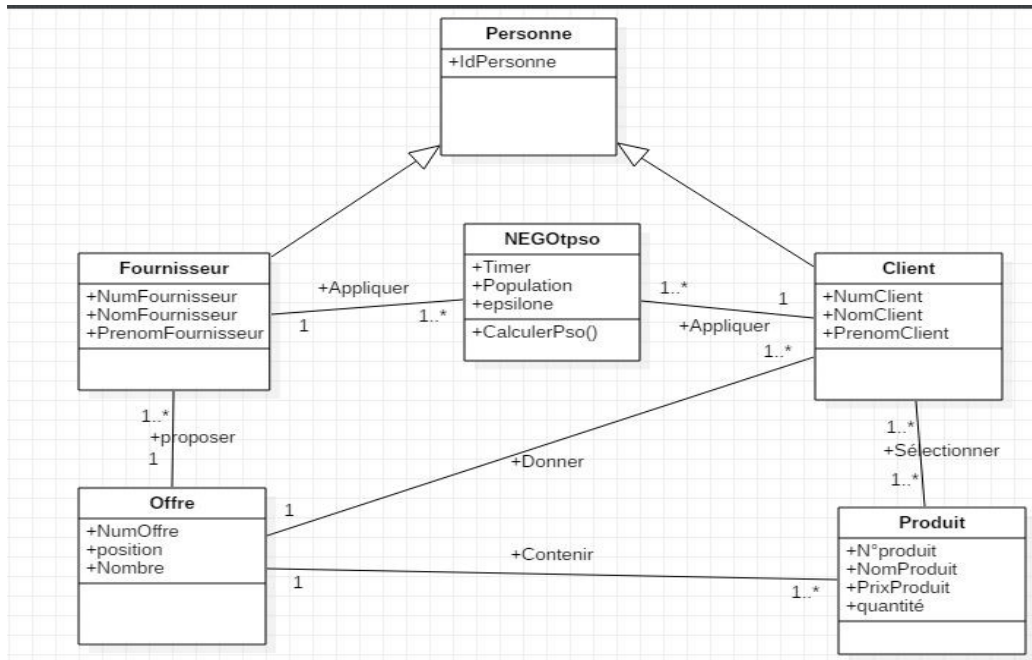


Figure 14 Diagramme de classe de négociation

Un cas d'utilisation est la description d'un ensemble de séquences d'actions qu'un système effectue pour produire un résultat observable à un acteur. Un cas d'utilisation représente une exigence fonctionnelle d'un système dans son ensemble. Les diagrammes de cas d'utilisation décrivent ce qu'un système fait du point de vue d'un observateur externe. L'accent est mis sur ce qu'un système fait mais pas sur la façon dont il le fait [44].

La figure suivante illustre représentant les rôles généraux de chaque service de notre système.

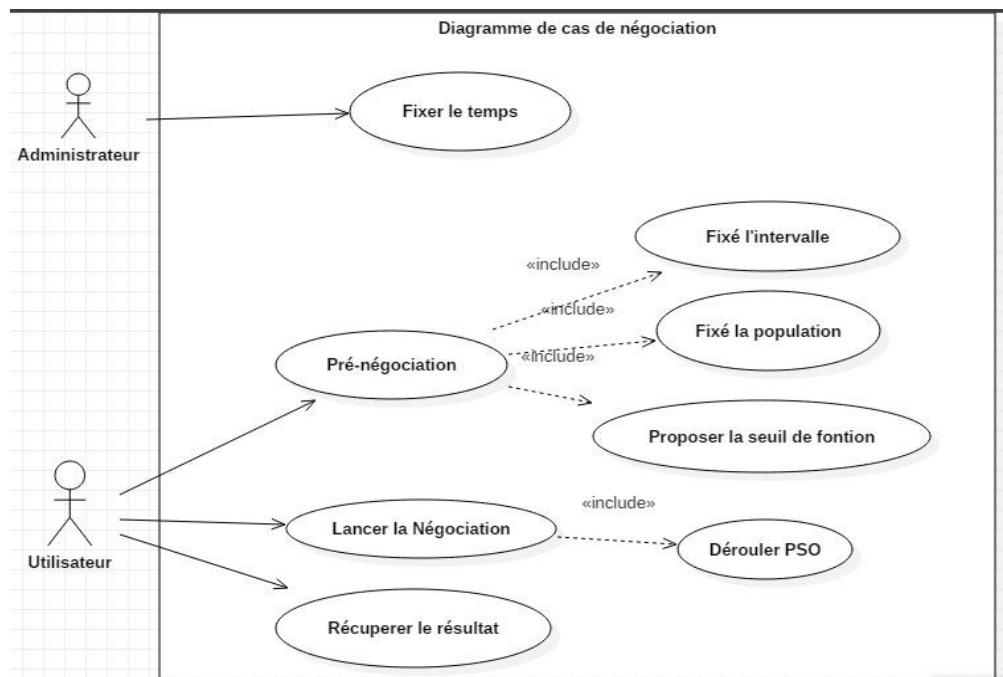


Figure 15 diagramme use case de négociation

D'abord, les utilisateurs vont commencer par la partie de pré-négociation. Chaque agent (client/fournisseur) doit proposer un intervalle et fixer une population limite le nombre des offres pour chacun d'eux. Ces utilisateurs vont donner ses fonctions qui s'appellent fonctions de seuil qu'elles représentent le pourcentage d'acceptation ou de rejet de l'offre.

Ensuite, la négociation va se lancer avec une durée précise par l'administrateur qu'elle doit contenir le déroulement de la méthode PSO pour chercher le résultat le plus performant pour les agents, en finissant par récupérer ce résultat obtenu.

7.2.2 Négociation à base de PSO et Tabou

La figure 16 représente les interactions des objets de notre système lors d'une négociation.

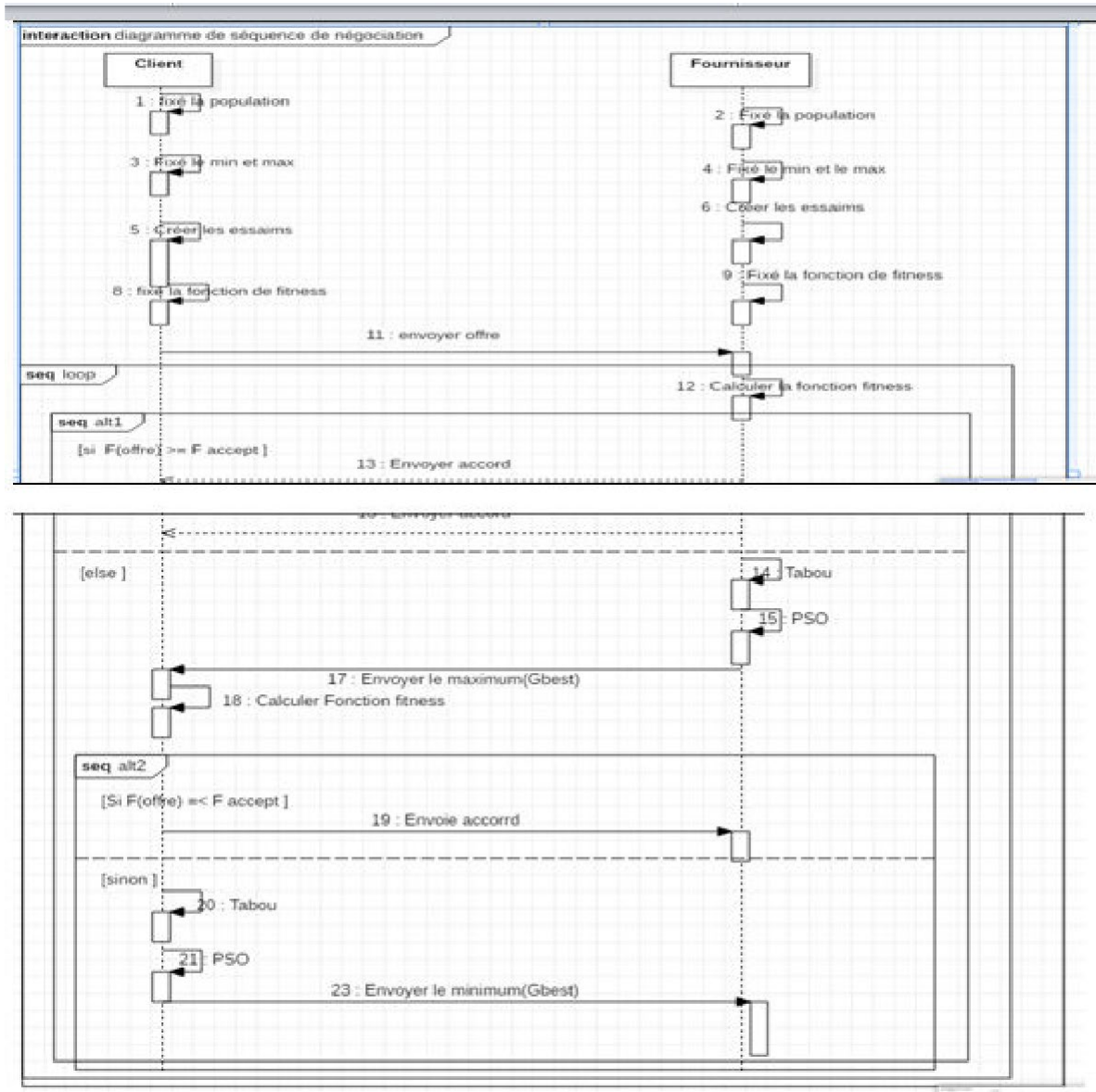


Figure 16 Diagramme de séquence de négociation

Afin de bien expliquer notre approche de négociation, nous avons utilisé l'ensemble des symboles illustrés dans le tableau 6.

X	Client/fournisseur.
O _i	Offre envoyée dans l'itération i .
E(O _i)	Ensemble des offres.
ε	la valeur de croisement des offres de tableau
N	Intervalle
Min	la valeur minimum d'intervalle
Max	la valeur maximum d'intervalle
S	seuil de fonction (pourcentage)
Q _o	qualité d'offre
P _{best}	la moyenne entre offre et ces voisins
G _{best}	la meilleure solution
V(t)	Le voisinage par rapport le temps
n	Le nombre de voisin par rapport le temps

Tableau 4: symbole de l'algorithme NEG_{TPSO},

Après la phase de pré-négociation, la procédure de négociation commence.

Chaque adversaire, envoie sa meilleure offre (O_i) à l'autre agent où est le nombre des itérations de négociation. Ce dernier (contre-adversaire) vérifié si cette offre (O_i) appartient à son intervalle (Min et Max) **c.à.d.**: $O_i \leq \text{Max}$ et $O_i \geq \text{Min}$.

Si l'offre est dans l'intervalle, une procédure de PSO est lancée :

- ❖ Calculer la fonction de fitness pour évaluer l'offre O_i.
- ❖ Si la valeur de la fonction de fitness n'est pas accepté ; elle sera ajouté au Tabou et l'agent calcule le nombre des voisins d'offre puis calcule le P_{best} et G_{best} (par rapport à l'essaim entier) puis envoie le G_{best} à l'adversaire.
- ❖ La procédure se répète jusqu'à :
 - ✓ Le temps de négociation termine (Timer =0).
 - ✓ La fonction de fitness n'offre reçus est acceptée (elle appartient au seuil désigné).

Pour ne pas revenir aux mêmes valeurs rejetées des offres envoyées, NEG_{TPSO} fait intervenir une liste T baptisée liste taboue. Cette liste permet de mémoriser durant un nombre limité d'itérations, les dernières solutions visitées et d'interdire tout déplacement vers ces solutions. On appelle toute solution figurant dans la liste T une solution taboue.

Dans les sections suivante : nous expliquons en détaille comment la fonction de fitness et le calcul de voisinage sont déroulés dans notre système.

a) Calculer la fonction fitness

- Dans le cas de Fournisseur : On doit commencer par calculer la distance (A) entre l'offre (O_i) et le Max tel que :

$$\text{Distance} = \text{Max} - O(i), \quad (A)$$

Comme la montre la figure 17 :

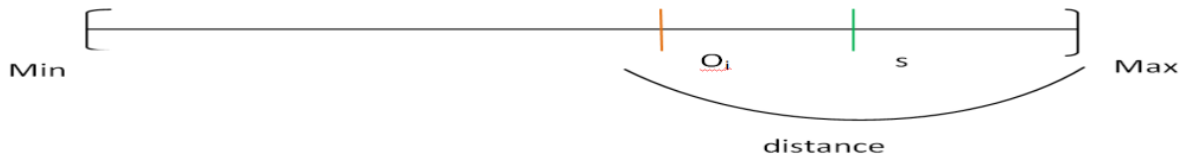


Figure 17 calculer la distance de fournisseur

- Calculer le pourcentage de distance

Si on considère que la distance : $\text{Dist} = \text{Max} - \text{Min}$ représente 100%, nous calculons la qualité de la position de l'offre O_i dans [min - max] par la fonction suivante :

$$\text{Ff}(O_i) = \frac{\text{Distance} * 100}{\text{Dist}} \quad (B)$$

- Dans notre cas une offre est acceptée si : $\text{Ff}(O_i) \geq S$. donc ici on annonce que la négociation est réussite.
- Si $\text{Ff}(O_i) < S$ l'offre sera refusée, donc l'offre (O_i) sera ajouté dans la liste Tabou (T) le fournisseur va passer à l'étape de voisinage (v).

Dans le cas de client, les étapes sont les même sauf que le calcule fera par rapport au Min. C'est à dire:

- Dans fonction de fitness, la distance devienne : $\text{distance} = O(i) - \text{Min}$.

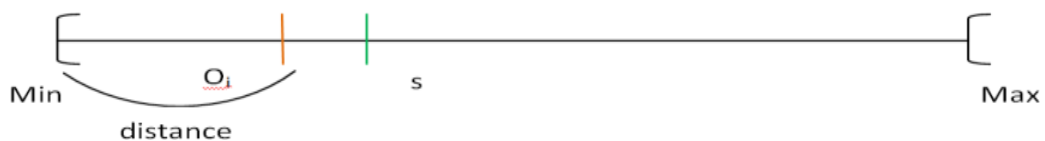


Figure 18 : calcule la distance dans l'état de client

- Dans le pourcentage de distance, une offre est acceptée si : $\text{Ff}(O_i) \leq S$.
- Dans le calcul de voisinage, après le calcul des pbest, le client doit sélectionner la meilleure valeur des nouvelles positions (Gbest) qui implique le minimum des pbest et l'envoyer au client qu'elle représente la nouvelle offre (O_i).

b) Calculer le voisinage & Pbest/Gbest

L'agent doit calculer la nouvelle position de l'offre (Pbest) par rapport au ses voisins dans chaque itération ou bien intervalle de temps. L'idée est d'augmenter le nombre des voisins pour le contre-offre si le timer diminue pour augmenter l'impact de contre-offre sur l'essaim sans accélérer le degré de renonciation. Dans ce cas, si le temps de négociation diminue, le nombre des voisins augmente.

Pour le calcul de Pbest, dans notre cas, nous avons utilisé un compromis entre la valeur d'offre et l'ensemble des voisins désignés. De cette manière, nous évitons le calcul lourd et nous accélérons la négociation.

Après le calcul de Pbest pour chaque particule dans l'essaim, un Gbest est envoyé à l'adversaire.

- ❖ Dans le cas de Fournisseur: Gbest est le maximum des offres dans l'essaim.
- ❖ Dans le cas de Client: Gbest est le minimum des offres dans l'essaim.

On résume toutes ces étapes précédentes par un diagramme d'activité comme elle montre la figure 19.

- Diagramme d'activité, associé à un objet particulier ou à un ensemble d'objets, qui illustre les flux entre activités. Il permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Les transitions sont déclenchées par la fin d'une activité et provoquent automatiquement le début immédiat d'une autre [43].

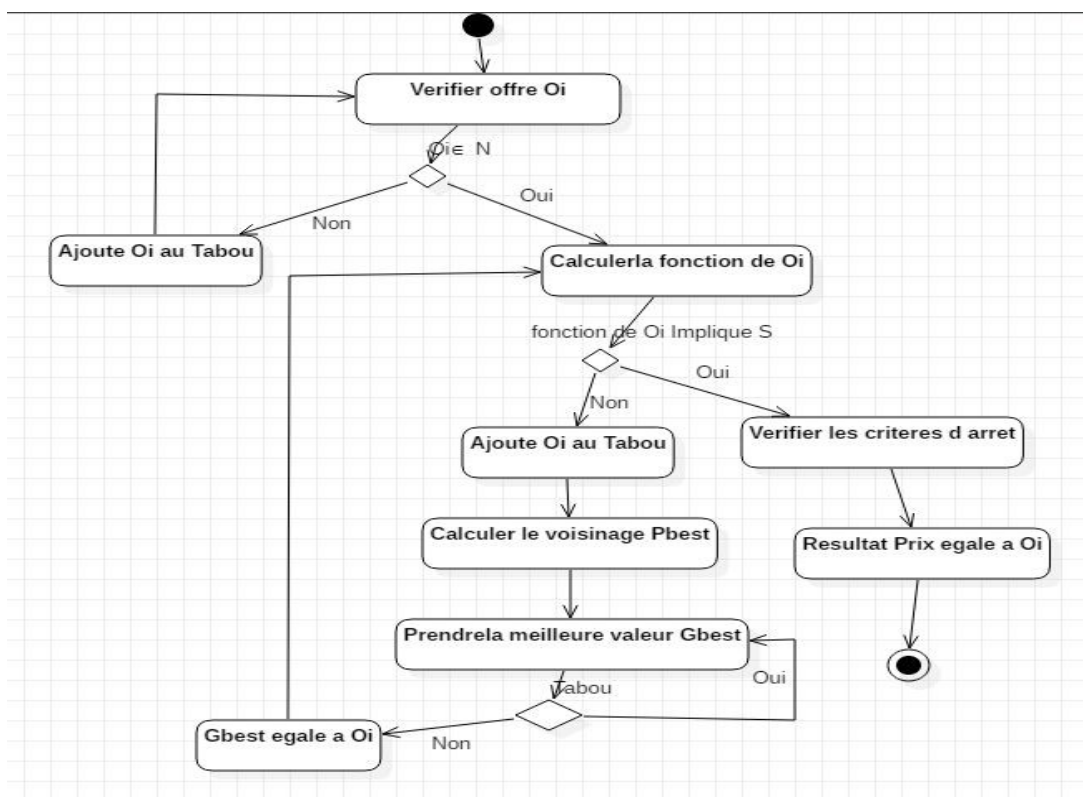


Figure 19 Diagramme d'activité de négociation

Soit :

Tableau 5: symbole de l'algorithme NEGOTPSO,

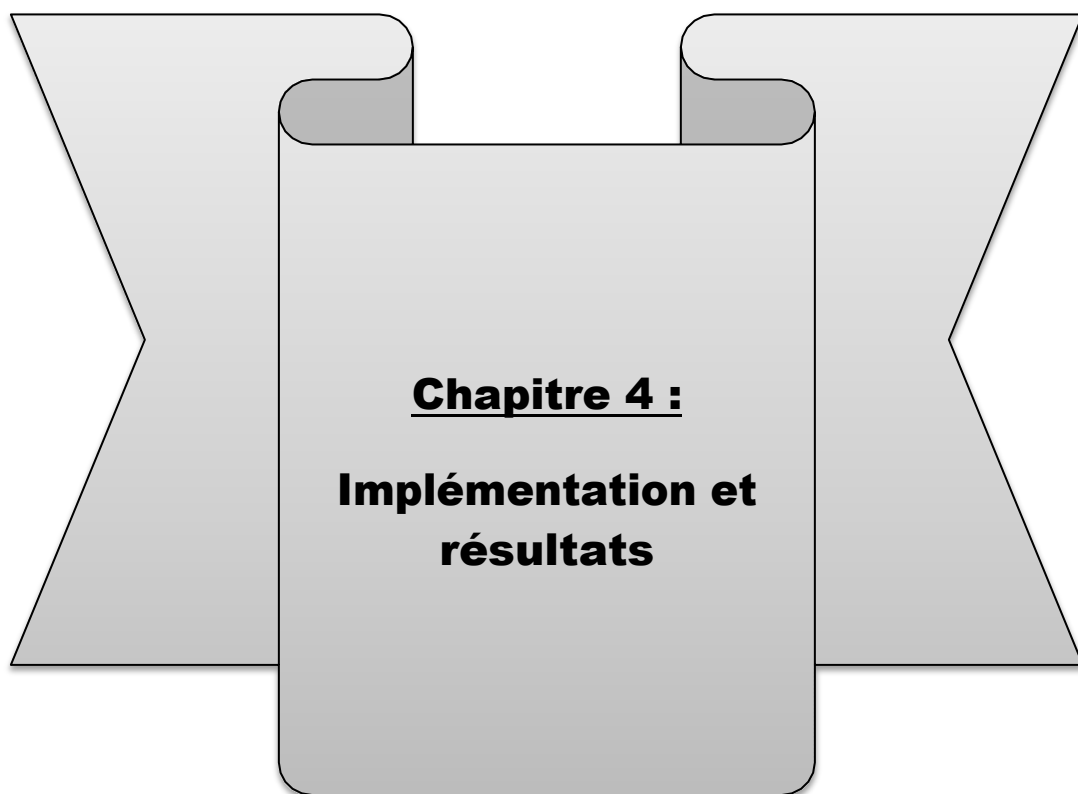
L'algorithme 3 résume toutes les étapes de négociation expliquées.

Algorithme3 : de notre approche NEGOTPSO,
Début
Fixé epsilon ϵ
Fixé le seuil de fitness S
Initialiser la population E(Oi) ;
Tant que (Oi \in N) faire
Calculer la fonction fitness ff(i)
Si Ff(i) accepte S
accepter offre
stop nego
Si non Ajouter Oi au tabou
Calculer le voisinage de Oi
Calculer pbest
Calculer gbest
Envoyer gbest
Fin Si
Fin Tant que
Fin

8 Conclusion

La phase conceptuelle est une étape fondamentale pour la réalisation de n'importe quel projet. Dans cette partie nous avons analysé la méthode de négociation à base de l'algorithme d'optimisation par essais de particule. Ensuite nous avons présenté les différents diagrammes d'UML pour expliquer notre approche proposé afin d'améliorer les limites de PSO.

Par la suite, nous allons montrer les aspects pratiques et la configuration des outils nécessaires liés à la réalisation et le fonctionnement de notre application.



1 Introduction :

Pour la conception de notre système, on a choisi UML car il s'adapte mieux pour le problème posé ainsi ça va nous faciliter la construction de l'application qu'on doit citer ces caractéristiques et ces parties en détails dans ce chapitre, et les outils utilisés pour la construction de notre application, on va parler alors de l'environnement de développement et les outils utilisés en commençant par le langage JAVA, suivi par la présentation de NetBeans qui est un environnement de développements (EDI), après on visualise les interfaces principaux de notre application.

2 Environnements de développement :

2.1 Environnement matériel

Pour développer cette application on a utilisé une machine, configurée comme suit :

- Machine ASUS sa mémoire Vive est de 8 GB, disque dur est de 500 Go et son processeur "Intel (R) Core (TM) i7-4500U CPU @ 1.80 GHz 2.40GHz " avec un système " Windows8 ". de type 64bit.

2.2 Environnement Logiciel :

Lors du développement de cette application, on a utilisé : Langage java (netbeans).

2.2.1 JAVA

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable [46].

Il est fourni avec un ensemble d'outils (le JDK Java Développement Kit) et un ensemble de packages : ensemble de classes. Ces différentes classes de base couvrent beaucoup de domaine (entrées/sorties, interface graphique, réseau, etc.) Cette richesse en "bibliothèques standards" explique sûrement en partie le succès de Java. Le langage lui-même se trouve dans le package java. Lang [47].

2.2.2 NETBEANS

NetBeans est un environnement de développement intégré (EDI), placé en Open Source par Sun. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, PHP, HTML ... Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris, Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java développement Kit (JDK) est requis pour les développements en Java. L'IDE Netbeans s'enrichit à l'aide de plugins. Afin d'installer correctement NetBeans, il est nécessaire d'installer le JDK compatible à la version de NetBeans [48].

2.2.3 JDK

Le Java Développement Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en byte code destiné à la machine virtuelle Java [49].

2.2.4 JADE

Le Java Agent DEvelopment framework (JADE) est un middleware écrit en Java et conforme aux spécifications de la FIPA. Cet environnement simplifie le développement d'agent en fournissant les services de base définis par la FIPA, ainsi qu'un ensemble d'outils pour le déploiement. La plateforme JADE peut être répartie sur un ensemble de machines et configurée à distance, grâce à un mécanisme de migration d'agent au sein de la même plateforme [50]

3 Les différentes interfaces graphiques :

Dans cette partie nous allons illustrer les différentes interfaces avec les traitements utilisés dans notre application.

➤ Page de démarrage

La figure (20) suivante représente la page de démarrage de notre application :



Figure 20 : Page de démarrage.

➤ Page d'accueil

La figure (21) ci-dessous représente la page d'accueil qui va contenir les différents produit que le client veut l'acheter.

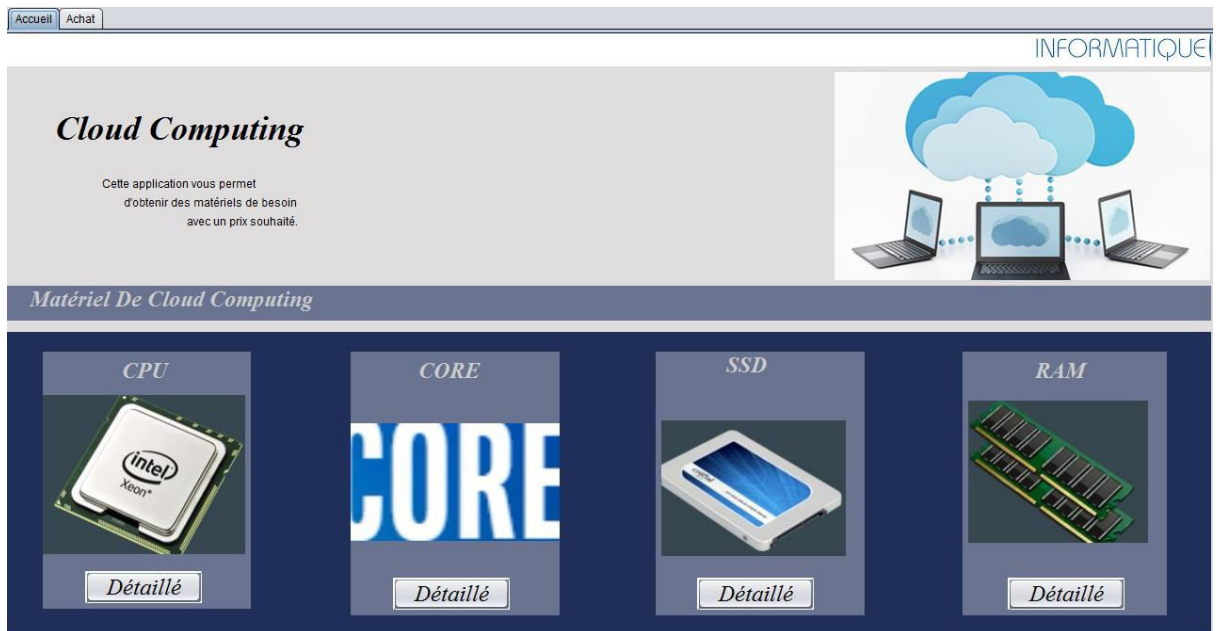


Figure 21: Page d'accueil

➤ Page d'achat

Si le client veut commander un produit, il va passer à la page d'achat qui va porter les différents détails qu'il a besoin. Comme elle représente la figure (22).

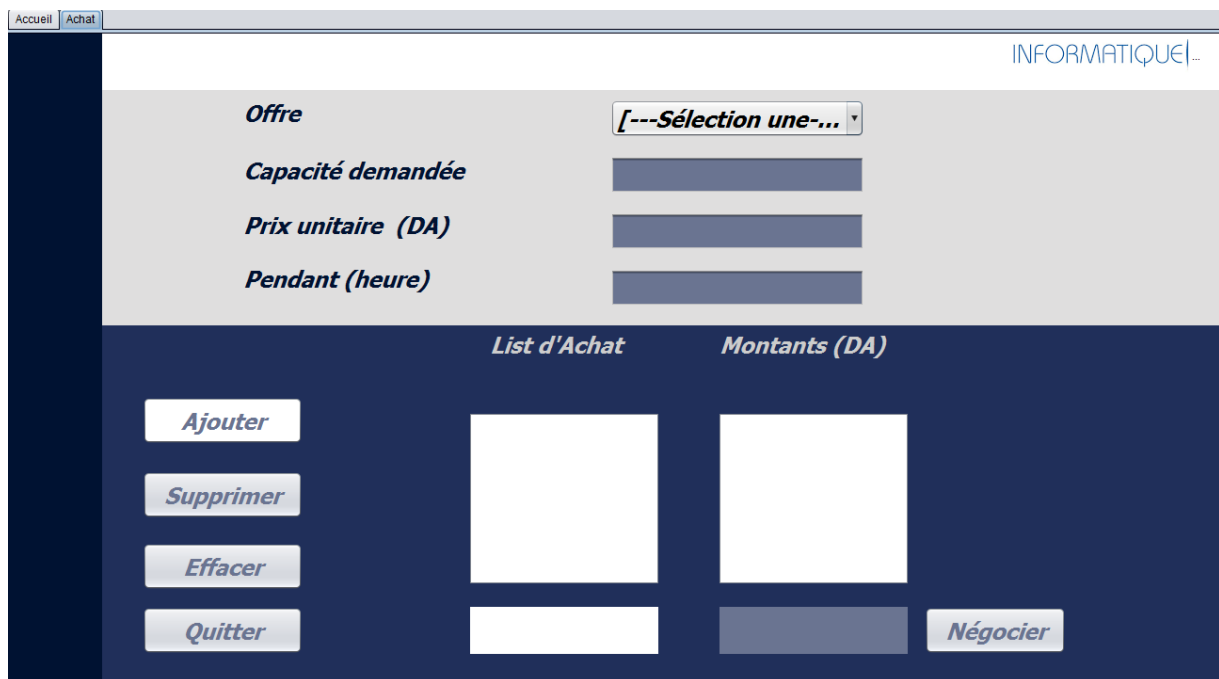


Figure 22: Fenêtre d'achat

Le client va sélectionner le produit qu'il va l'acheter, et précise sa capacité et le temps qu'il a besoin comme elle montre la figure (23) :

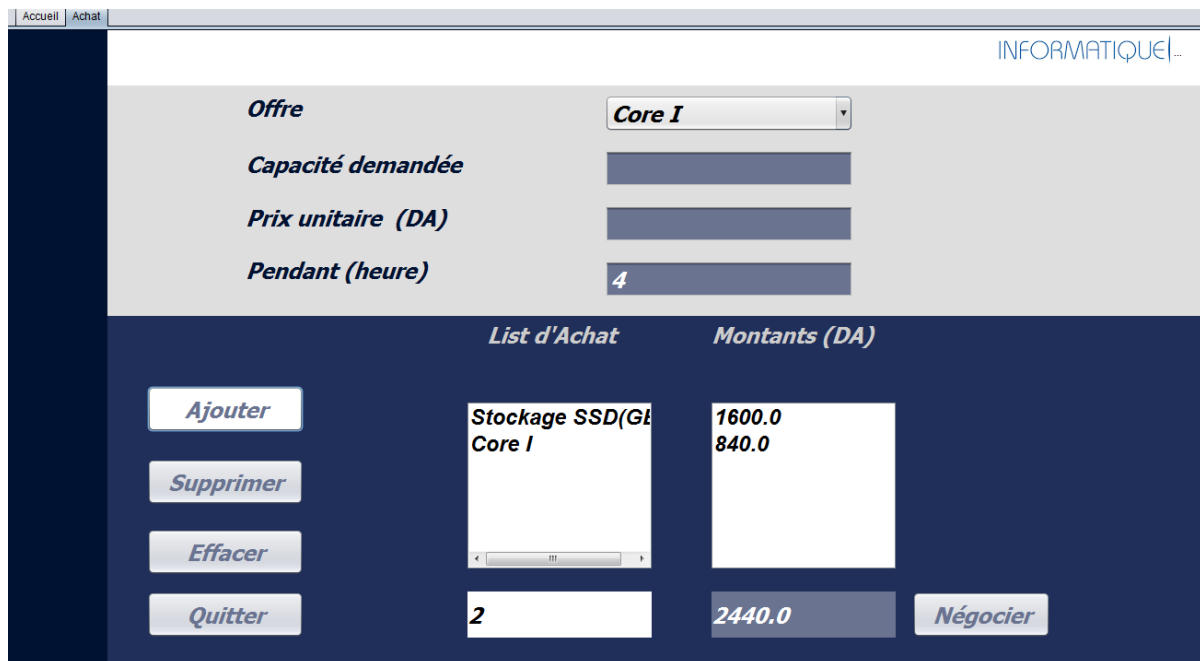


Figure 23: Sélection des produits.

Quand le client clique sur le bouton « ajouter » sans remplir un des cases trouvées, il va s'afficher un message d'erreur :

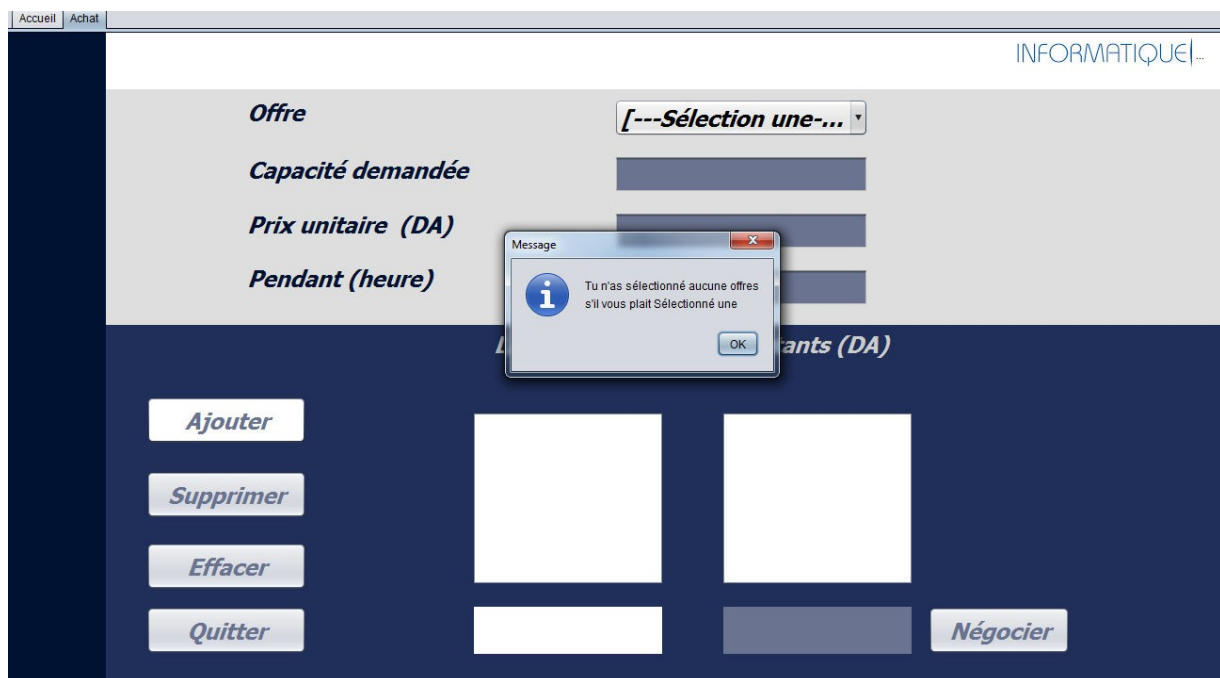


Figure 24: Message d'erreurs.

➤ Page de négociation de client

La figure(25) représente la page des paramètres de client :



Figure 25: Page de négociation de client.

La figure (26) montre le tableau des offres obtenu après que le client remplis ses paramètre :

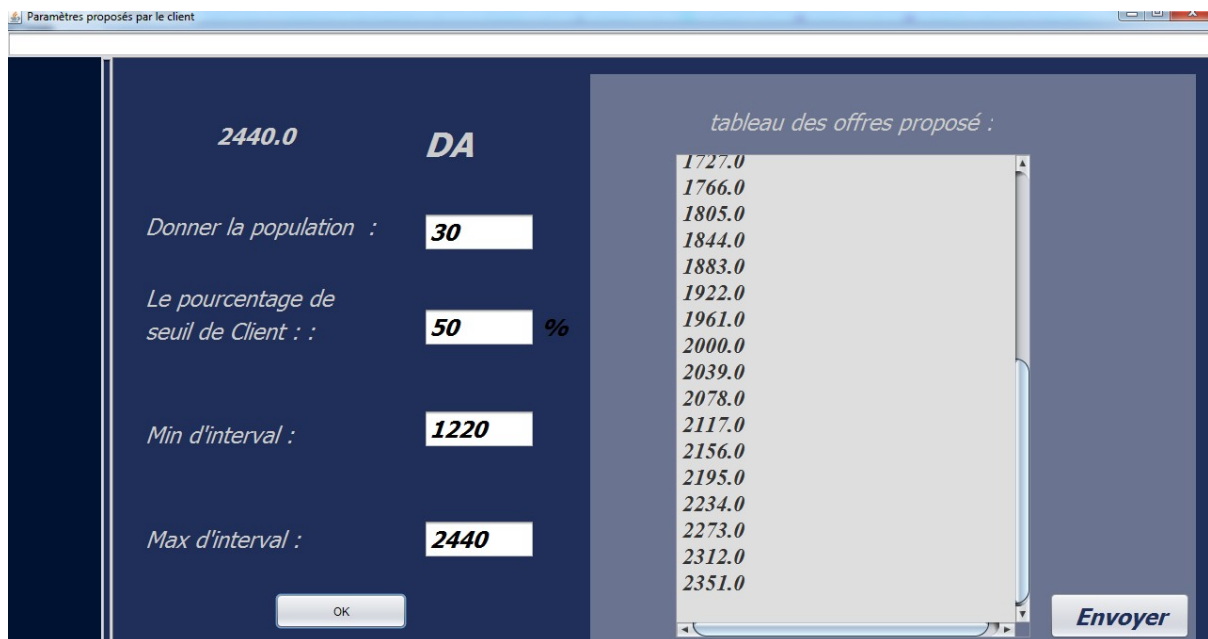


Figure 26 : Paramètre donnée par le client.

➤ Page de négociation de fournisseur

La figure (27) illustre les paramètres de fournisseur qu'il va saisir :

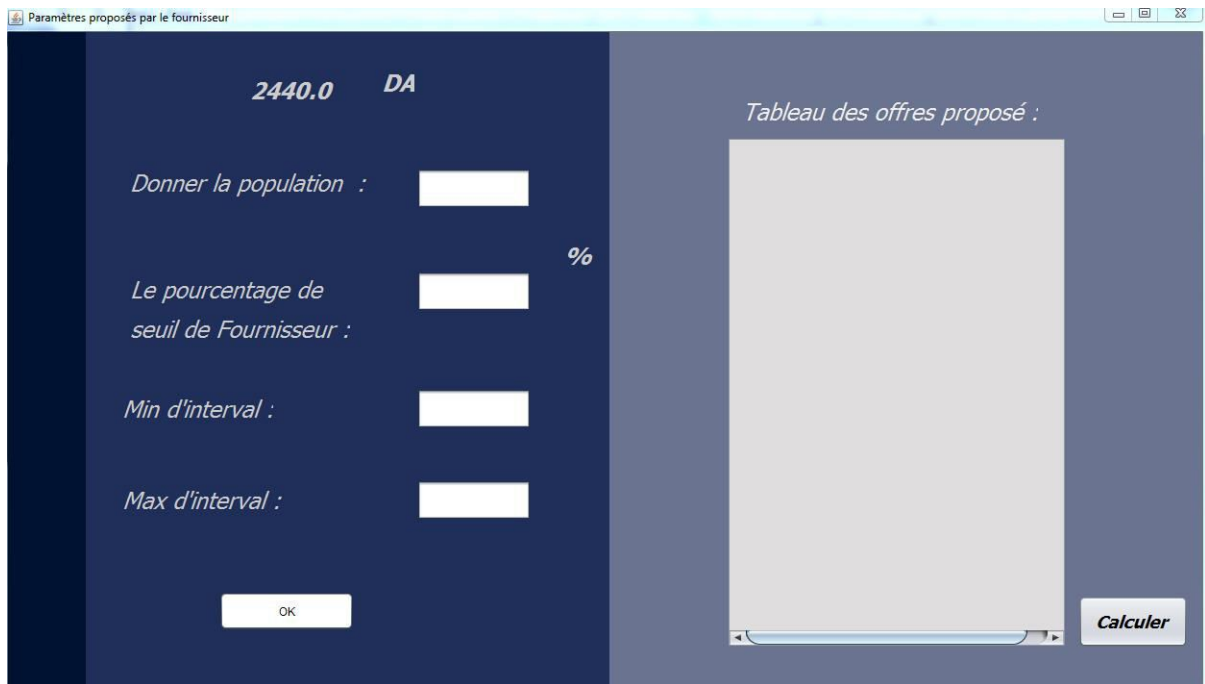


Figure 27:Page de négociation de fournisseur.

Après que le fournisseur saisie ses paramètre, on va obtenu le tableau de ses offres (figure 28) :

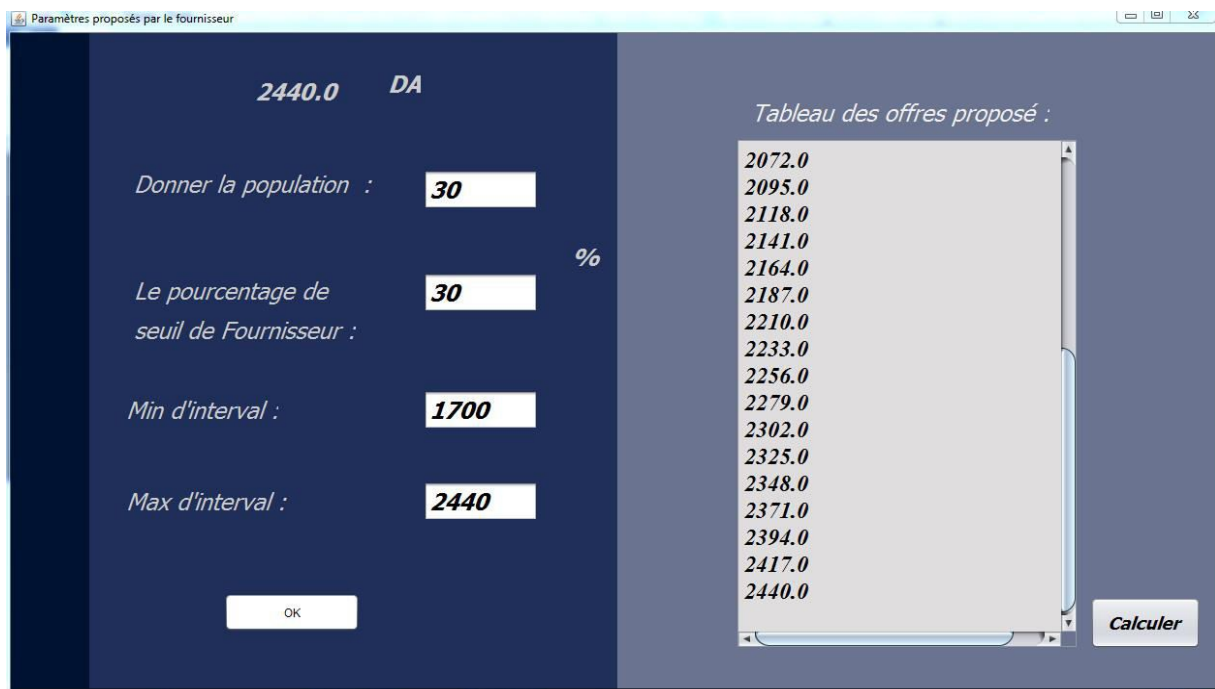


Figure 28: Paramètre données par le fournisseur.

➤ Page de résultat final

Après que le client et le fournisseur saisissent leurs paramètre, il ont passé à ma page final des résultats figure (29) :

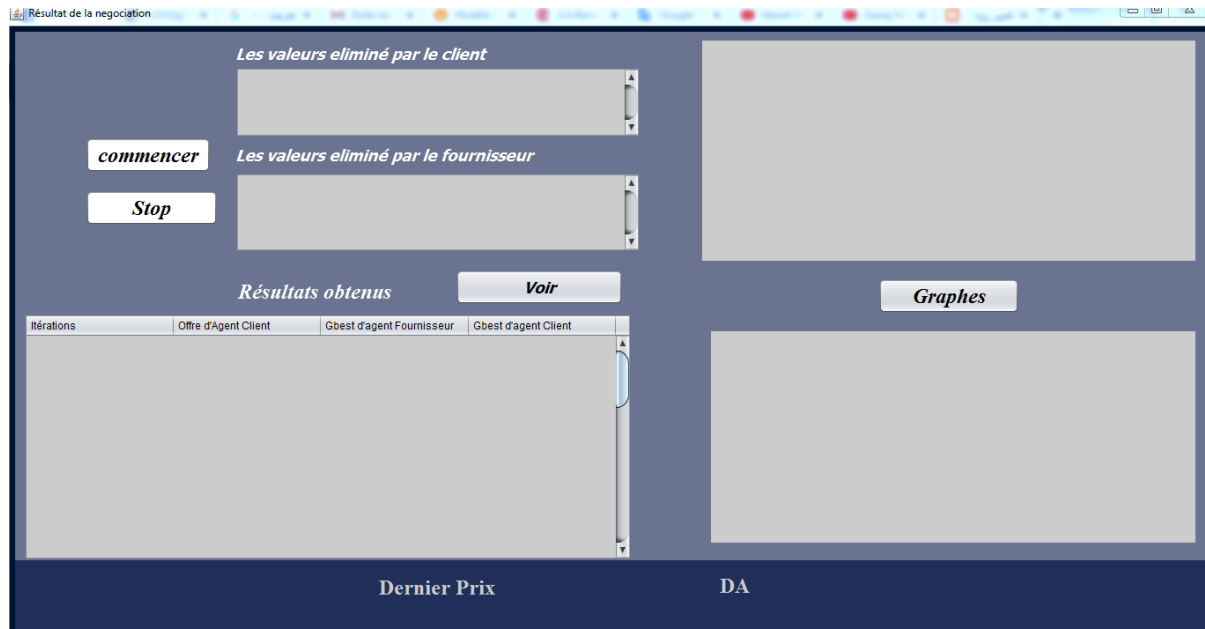


Figure 29: Page des résultats finals

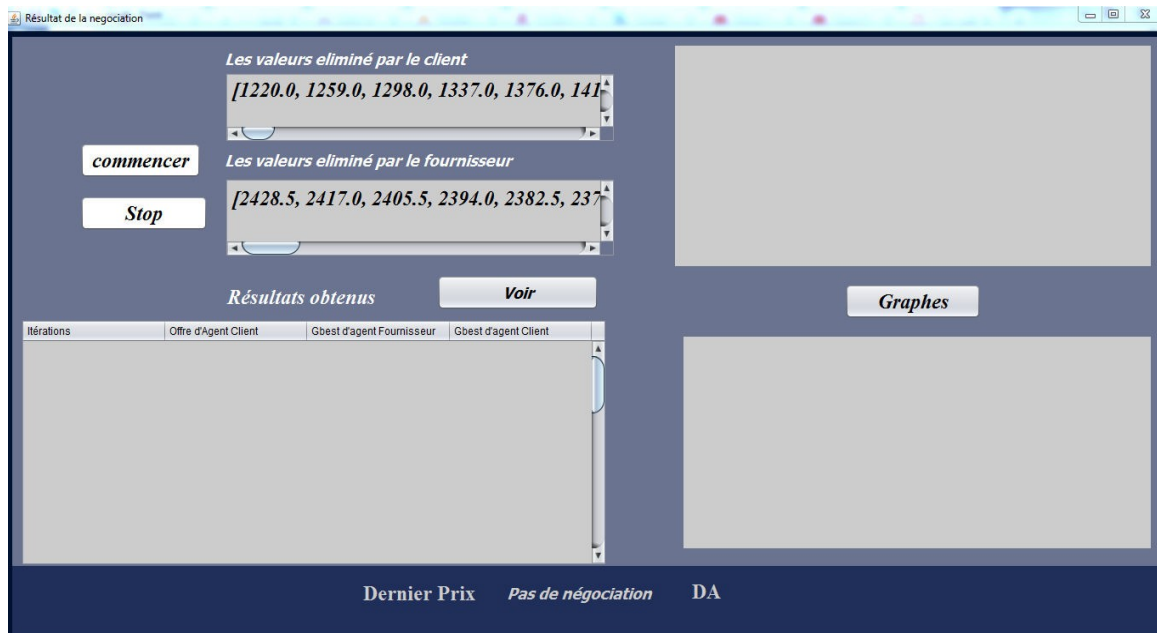


Figure 30: Résultat de la liste de tabou de client et de fournisseur

La figure (30) représente la liste éliminatoire des offres pour le client et le fournisseur : elles sont des listes des offres refusée.

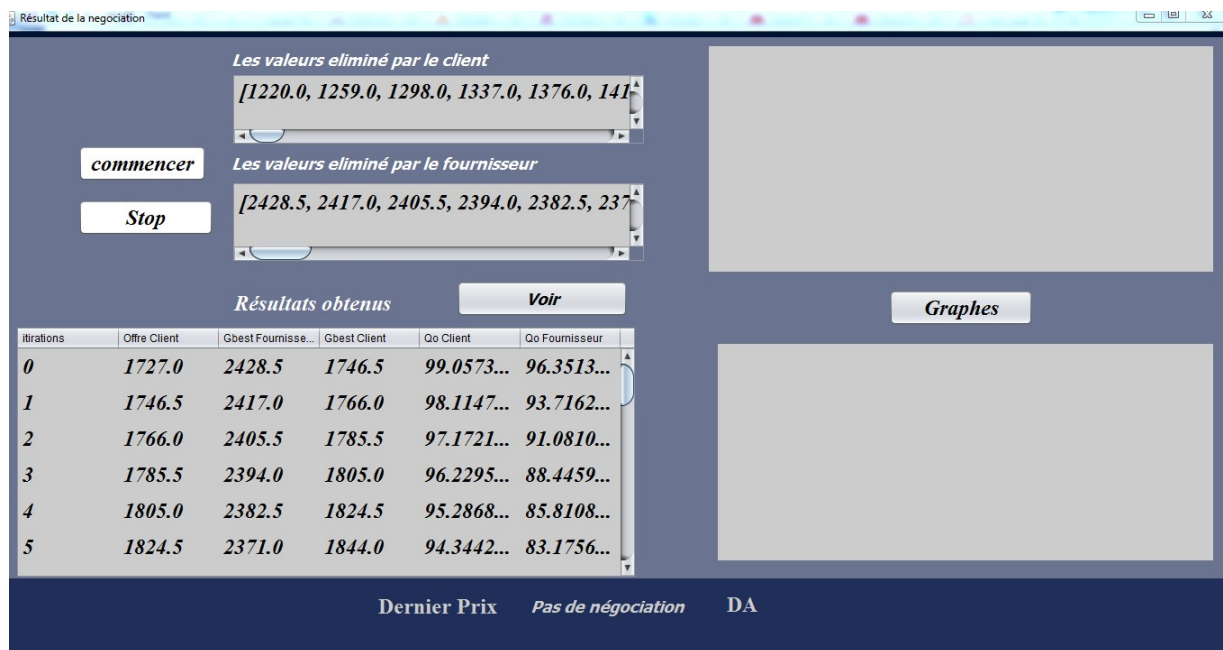


Figure 31: Résultats obtenus d'un exemple.

Cette page ci-dessus (figure 31) possède les résultats obtenus lors de la simulation de notre négociation d'un exemple traiter avec un graphe des itérations comme elle montre la figure (32).

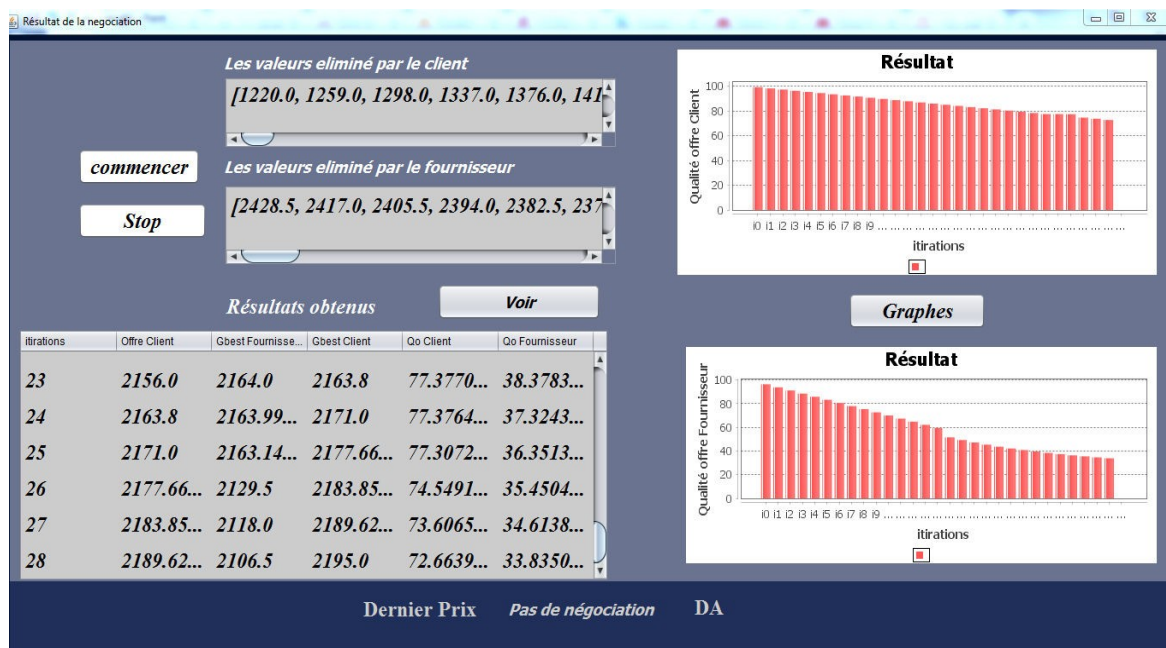


Figure 32: Affichage des graphes des résultats de chaque itérations.

4 L'analyse des résultats

- **NEGO_p VS NEGOT_{PSO}**

Ce schéma (figure33) représente la comparaison des résultats de la qualité d'offre (Pourcentage) par rapport aux itérations des temps entre les résultats de NEGOp (sans tabou) et NEGOT_{PSO} (Avec tabou).

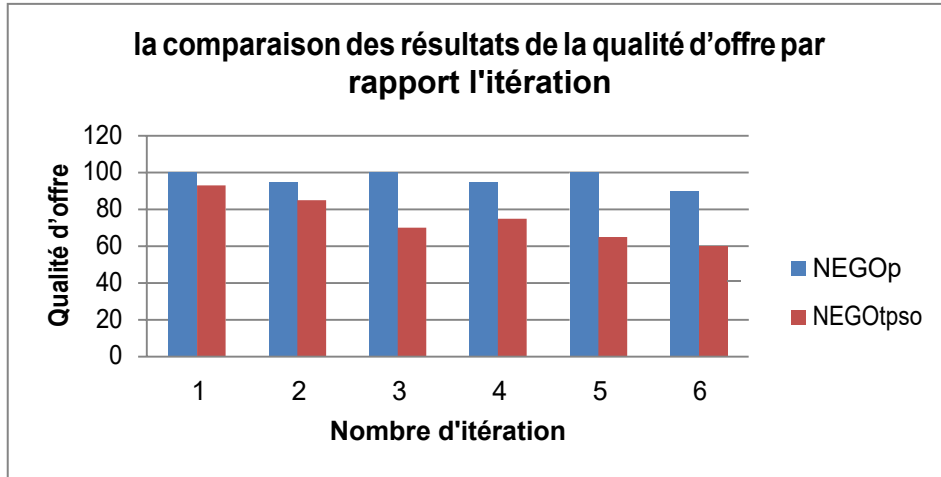


Figure 33 : Résultat entre NEGOp et NEGOTps0.

D'après le teste des résultats, nous avons remarqué que la valeur de changement des offres dans NEGOp est presque la même et cela rend difficile de trouver la solution optimale. Et dans des certains cas, le déroulement revient a les même valeurs donc on va trouver un perdre du temps avec une difficulté d'afficher l'offre satisfaisaire.

Par contre, NEGOT_{PSO} a éliminé les valeurs utilisée dans des itérations précédentes pour faciliter les calculs et ne pas revenir au même cas refusé. Comme on a vu que la qualité d'offre a changé lors de traitements et elle diminue temps en temps jusqu'à obtenir le résultat requis.

- **Nombre de succès parrapot au nombre de population**

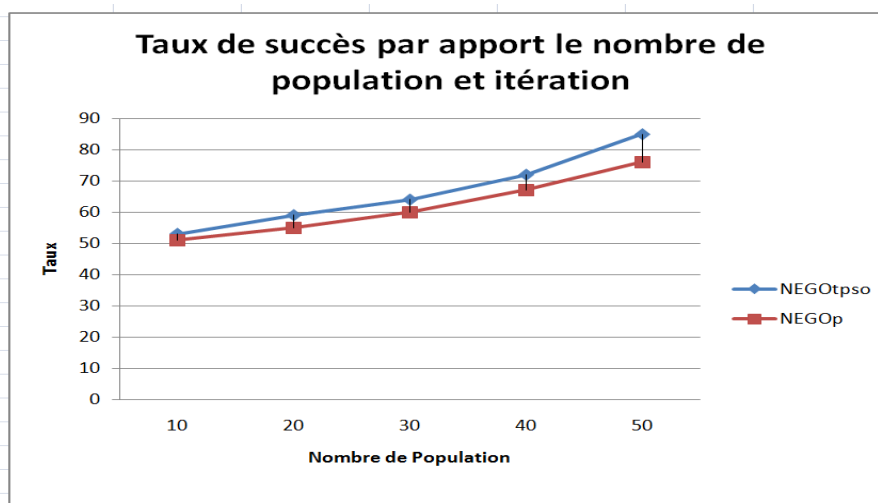


Figure 34 : Nombre de sucée avec la population

Dans cette expérimentation nous avons fait varier le nombre des offres et nous avons mesuré le taux de succès à chaque fois les résultats illustré dans la figure (34) l'augmentation du nombre des offres fait augmenter le taux de succès des deux algorithmes NEGOTPSO et NEGOp car le nombre des occasions augmente. Ce qui influence le taux de succès.

La différence de l'augmentation entre NEGOp et NEGOTPSO a suivi le changement des critères d'arrêts entre eux ce qui explique les bons résultats par rapport l'NEGOp.

- **Degré de satisfaction entre l'approche (NEGOp) et (NEGOTPSO)**

La figure (35) illustre le degré de satisfaction des offres entre NEGOp et NEGOTPSO.

Lors d'avancement de NEGOp, le degré de renonciation n'assure pas un degré de satisfaction élevé car le traitement va se dérouler de manière irrégulière qui utilise les ressources de façon illogique ce qui minimise la possibilité de satisfaire la demande de client dans un temps donné.

Par contre, la renonciation de NEGOTPSO est augmentée par chaque itération car elle utilise les ressources de façon régulière logique c'est pourquoi il trouve une solution demandé adéquate au client.

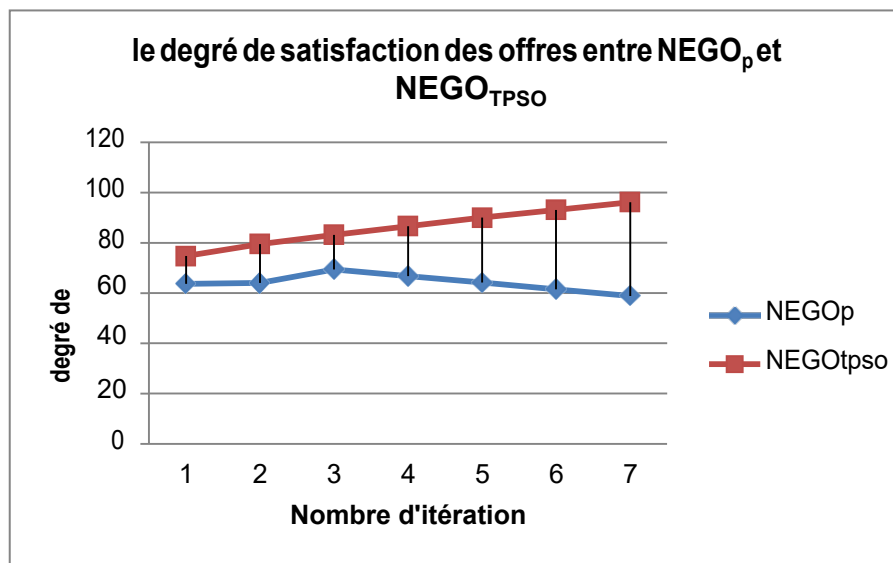


Figure 35 : degré de succès.

5 Conclusion

Dans ce dernier chapitre, nous avons présenté les outils utilisés pour réaliser notre projet ainsi que les différentes interfaces permettant d'accéder aux différentes fonctionnalités de notre application de Gestion négociation à base de PSO.

VI. Conclusion générale

Aujourd'hui, la négociation est au centre des préoccupations majeures au sein de toute entreprise. La raison de cet intérêt porté à la négociation en entreprise est toute simple : le développement et la performance de toute entreprise reposent sur une réelle négociation de celle-ci, avec ses clients et son personnel que cela soit de type commercial ou social.

Puisque le conflit est toujours une épreuve difficile que l'on souhaite éviter, la négociation fournit un moyen aux agents pour le résoudre, en leur permettant de trouver un accord mutuellement acceptable, elle représente un outil puissant pour régler les disparités entre agents. C'est un mécanisme qui permet aux agents de proposer des offres en explorant la possibilité de réaliser des buts à moindre coût.

Dans notre travail, on a étudié des méthodes utilisées dans la négociation : théorie des jeux qui classe les jeux en catégories en fonction de leurs approches de résolution, la méthode heuristique qui permet de fournir une solution plus proche de l'optimal et la méthode de l'argumentation permet aux agents de donner des informations supplémentaires dans leurs propositions.

La négociation dans le cadre de SLA a une particularité spéciale, elle est généralement contrôlée par le temps et la disponibilité de ressources :

- Une négociation qui favorise les négociateurs généralement ne respecte pas le temps.
- Une négociation qui ne considère que le temps peut causer un degré de renonciation rapide d'un ou de deux parties de négociation.

Afin de résoudre ces deux problèmes, nous avons proposé une stratégie de négociation bio-inspiré (NEGOTPSO) qui combine entre deux méthodes d'optimisation connues : Tabou et PSO.

On a expliqué en détail toutes les étapes de notre algorithmes , puis nous avons montré les phases nécessaires à la réalisation de cet application ce qui nous a appris à mieux manipuler le langage JAVA et approfondir nos connaissances sur l'utilisation de plateforme JADE.

Cependant des perspectives d'améliorations de notre application restent envisageables, parmi d'eux :

- Nous souhaitons d'étendre NEGOTPSO pour supporter la négociation collective (multifournisseur et multi client).
- Nous pensons aussi à intégrer NEGOTPSO au simulateur CloudSim pour fournir directement les prix adéquats pour le contrat SLA.

Bibliographie

- [1] : Document: Torry Harris, Cloud Computing.
- [2] : Mémoire de Master fin d'étude : MAIOUA Khalifa MANSOURI Adel, Approche basée Agents Mobiles intelligents dans un environnement de Cloud Computing, UNIVERSITE KASDI MERBAH OUARGLA, 2013/2014
- [3] : Liezel Cilliers, Types of Services offered by cloud computing
- [4] d'André Jeannerot : CE QU'IL FAUT SAVOIR SUR LE CLOUD COMPUTING, 2012.
- [5] : Abdelmajid, Les différents types de cloud computing, 2019.
- 5.1 [6] : Fernand-Valéry LEBORGNE, Université international SUPINFO, Cloud Computing, 2016.
- [7] : Bastien L, Cloud Computing- définition, avantage et exemples d'utilisation, 2017.
- [8] : Dufour Laurent, Qu'est Ce Que Le Cloud Computing ?, 2014.
- [9] : Robert Jaques, RJConsulting Sarl. SLA – Service Level Agreement Marche à suivre
- [10] : Saad Mubeen, Sara Abbaspour Asadollah, Alessandro Vittorio Papadopoulos, Mohammad Ashjaei, Hongyu Pei-Breivold, Moris Behnam , Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study, 2018.
- [11]: Mikael DEMETTE, Qu'est-ce que le SLA ? (Service Level Agreement), PROMOSOFT Informatique, 2014.
- [12] : Yousri.Kouki, Thomas.Ledoux, SLA et qualité de service pour le Cloud Computing, Université de Grenoble France, 2013.
- [12] : AXELOS , Common glossary of terms and definitions, October 2012.
- [13]: Liliane Bahali Bugeni, La négociation dans le commerce électronique, 1999-2000.
- [14] : l'Équipe Perspective monde, outil pédagogique des grandes tendances mondiales depuis, NEGOCIATION, 1945.
- [15] : Mme Riad Amel, Mémoire de magister (développement d'une architecture Multi-agent pour un système interactif d'aide multicritère à la décision (SIAMD) en gestion de production, Université d'Oran, 2009.
- [16] : Mourad Sassi, Mémoire (Protocoles de planification et de négociation dans les domaines orientés tâches) la Faculté des études supérieures de l'Université Laval.
- [17]: Kersten E.G, Negotiation Processes. Electronic Negotiation Media and Electronic transaction, 2002.
- [18]: LES DIFFERENTS TYPES DE NEGOCIATIONS Synthèses de réflexions de l'A.P.M.

- 6 [19]: René Oproiu, Négociation distributive, 13 juin 2014.
- [20]: Philippe Duvocelle, La négociation intégrative, 2014.
- 7 [21] : Juliane Chainière, Types de négociation, 29 octobre 2013.
- [22]: Al Falasi, Asma & Serhani, Mohamed & Hamdouch, Younes. A Game Theory Based Automated SLA Negotiation Model for Confined Federated Clouds. 10.1007/978-3-319-38904-2_11. En 2015.
- 8 [23] : Shyam, Gopal Kirshna, Manvi, Sunilkumar S, Prasad, Bhanu; Concurrent and cooperative negotiation of resources in cloud computing: A game theory based approach, Journal: Multi-agent and Grid Systems, vol. 14, no. 2, pp. 177-202, 2018.
- [24] : Xianrong Zheng, Patrick Martin, Kathryn Brohman, Cloud Service Negotiation: Concession vs. Tradeoff Approaches, 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.
- [25] : Georgiana Copil, Daniel Moldovan, Ioan Salomie, Tudor Cioara, Ionut Anghel, Diana Borza "Cloud SLA Negotiation for Energy Saving - A Particle Swarm Optimization Approach", 8th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP2011), Cluj-Napoca, Romania, Page(s): 289-296, 2012.
- [26] : Huang S. et Lin C.Y., 2010. The search for potentially interesting products in an emarketplace: An agent-to-agent argumentation approach. Expert Systems with Applications, 37 (6) : 4468–4478.
- 9 [27] : Roy McMurtry, Brian Dickson, Le manuel relatif au règlement des conflits, 2016.
- [28] : JAWAD Benabdallah, LE RÈGLEMENT PACIFIQUE DES DIFFERENDS INTERNATIONAUX, Université Mohamed 1er Oujda.
- [29] : MERABTI Hocine, Approches bio-inspirées pour la reconnaissance de formes, Université 8 Mai 1945 Guelma, 2017.
- [30] : Mohamed SANDELI, Traitement d'images par des approches bio-inspirées Application à la segmentation d'images, Université Constantine 2, 2013-2014.
- [31] : Moussa SEMCHEDINE, Contribution à la segmentation d'images médicales par les algorithmes bio-inspirés, Université Ferhat Abbas – Sétif 1-, 2018.
- [32] : Deneche Abdelhakim, Approches bio-inspirées pour la reconnaissance de formes, UNIVERSITE MENTOURI DE CONSTANTINE, 2006.
- [33] : ZEKRI Meriem, Approches Bio-inspirées pour la Fouille de Données en Bio-informatique, UNIVERSITE BADJI MOKHTAR-ANNABA, 2014-2015.
- [34] : Christelle REYNES, Etude des Algorithmes Génétiques et Application aux Données de protéomique, UNIVERSITE MONTPELLIER I, 20 juin 2007.

[35]: J. Kennedy and R. C. Eberhart. "Particle Swarm Optimization". In : Proceedings of the IEEE International Conference on Neural Networks IV, November 1995.

[36]: Tong Xiangrong, Zhang Wei, **Agent negotiation over multiple divisible resources with nonlinear utility functions under incomplete information**, International Conference on Systems and Informatics (ICSAI), Page(s):674-677, 2012.

[37]: Craig W. Reynolds, Flocks, herds, and schools: A distributed behavioral model, Computer Graphics, 1987.

[38]: S. Jaya Nirmla, Shah Maulik, S. Mary Saira Bhanu, **SLA achievement by negotiation in a cloud**, Proceeding Compute '13 Proceedings of the 6th ACM India Computing Convention, 2014.

[39] : Mr BESTAOUI Abdallah Anes , Gestion de spectre dans un réseau de radio cognitive en utilisant l'algorithme d'optimisation par essaim de particules, Université Abou Bakr Belkaid– Tlemcen , 2014 – 2015

[40] : Maxime BOMBRUN Abdoulaye , L'optimisation par essaim particulaire pour des problèmes d'ordonnancement , Rapport d'ingénieur Projet de 2ème année ,24/03/2011.

[41] : CLERC et SIARRY, Une nouvelle méta heuristique pour l'optimisation difficile : la méthode des essaims particulaires, 2003.

[42] : EUDES et RIOLAND, Optimisation par essaim particulaire pour un problème d'ordonnancement et d'affectation de ressources, 2007.

[43] : Maria Zemzami , Norelislam Elhami , Abderahman Makhloufi , Mhamed Itmi , Nabil Hmina , Application d'un modèle parallèle de la méthode PSO au problème de transport d'électricité , ISTE OpenScience – Published by ISTE Science Publishing, London, 2016.

[44]: Spécialisation Sciences Cognitives et Informatique Avancée, Optimisation par essaim de particules, 94270 Le Kremlin-Bicêtre, France.

[45]: Slimene Ranim & Selmi Sabrina, Application Web de Gestion de stock du magasin de Faculté de Médecine, Université de Monastir, 03/07/2012.

[46] : BELGACEM Hicham. - BOUAZZA Mohammed Reda, Gestion d'une agence de voyage selon le modèle Client/Serveur avec Swing et MySQL, Université Abou Bakr Belkaid–Tlemcen, 09 Juin 2014.

[47]: diplôme de magister : Kahloula Boubaker, Chargement de donnée XML dans un data Warehouse :Approche pour l'automatisation de schéma Matching, université d'Oran, 2009/2010.

[48] : Olivier CHATOR, ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGENIEUR, L'UNIVERSITÉ de BORDEAUX, 27 Mars 2015.

[49]: Obtention de licence : DAKHLI Amira et MATAR BACHA Malek, Conception et Développement d'une application de gestion d'une base de connaissances au sein de la CNAM, université virtuel de Tunis, 2017.

[50] : L'obtention du diplôme Master : Mr BELHOUD Fatah. Mr BELMEHDI Rafik, Mise en œuvre d'une approche de composition de services web à base d'agents logiciels, Université A/Mira de Béjaia, 2015/2016.