

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE**



**UNIVERSITE ABOU BAKR BELKAÏD – TLEMEN**



**FACULTE DES SCIENCES**

**DEPARTEMENT D'INFORMATIQUE**

**THESE**

Présentée par

**Benamar Abdeladim**

**Ait Hamouda Mounir**

Pour obtenir le diplôme de  
**MASTER EN INFORMATIQUE**

Spécialité : *Modèle Intelligent et Décision(M.I.D)*

Sur le thème

**Sélection de services web par l'Optimisation d'Essaim  
Particulaire hybride**

**Soutenue publiquement le 15 / 06 /2013, devant le Jury composé de**

<b>Président</b>	Mr Benmammar badr	U. Tlemcen
<b>Directeur de Thèse</b>	Mr. HADJILA Fethallah	U. Tlemcen
<b>Examineurs</b>	Mr. Merzoug mohammed	U. Tlemcen
	Mr. BELABED Amine	U. Tlemcen

Année universitaire : 2012-2013

# REMERCIEMENTS

Nous tenons à remercier :

Allah le tout puissant.

Mr HADJILA FETHALLAH, notre encadreur, pour ses conseils, sa disponibilité et son encouragement qui nous ont permis de réaliser ce travail dans les meilleures conditions.

Les jurys pour leurs efforts et leur soin apporté à notre travail :

Mr. Benmammar badr

Mr. Merzoug Mohammed

Mr. BELABED Amine

Aux enseignants de notre université et du département d'informatique et notre collègue de promotion.

## DEDICACE

*Je dédie ce mémoire :*

- *A Ceux qui ont fait de moi l'homme que je suis aujourd'hui : mes très chers parents : ma mère, ma mère, ma mère et à mon père, que dieu les récompense et les garde, et surtout ma mère qui m'a éclairée mon chemin et qui m'a encouragé et soutenue toute au long de mes études.*
- *A Mon frère Ammar et ma sœur Wissem.*
- *A mes cousins et cousines.*
- *Spécial dédicace à mes collègues Ayoub et Fouzi et les 4 amis les M.CH (Hakim, Oussama, Yasser et mustapha) de SIC.*
- *A tous les membres de ma promotion de MID.*
- *A mes amis sans exception.*
- *A tous mes professeurs.*

*Benamar Abdeladim*

## DEDICACE

*Je dédie ce mémoire :*

*Louange à Dieu, Seigneur des mondes ;*

*C'est toi que nous adorons et de toi que nous implorons secours ;*

*Que la paix et la bénédiction soit sur son dernier envoyé ;*

*A mes chers parents qui ont été présents, à chaque instant, Leur  
irremplaçable et inconditionnel soutien m'a permis d'écarter les*

*doutes, de soigner les blessures et de partager mes joies ;*

*A ma petite famille, mes frères et sœurs ;*

*A tous mes oncles et tantes ;*

*A mes cousins et cousines ;*

*A tout la famille Ait Hamouda et Haballah ;*

*A tous mes amis sans exception.*

*Ait Hamouda Mounir*

## Résumé

Le nombre de services web est en augmentation constante sur le réseau internet, par conséquent ; les utilisateurs ont besoin de solliciter des approches qui leurs permettent d'effectuer la sélection voulue, surtout dans le cas des compositions de services (complexes). Ces approches doivent prendre en compte les différents attributs de QoS offerts par les fournisseurs, de plus elles doivent satisfaire les exigences globales de l'utilisateur (tels que le coût total des compositions).

Pour assurer ces objectifs, nous proposons dans ce travail une méta-heuristique hybride qui combine les essaims particulaires avec les opérateurs de croisements évolutionnaires. Les résultats obtenus sont très encourageants et confirment l'utilité des méta-heuristiques dans ce genre de problèmes.

**Mots clés :** services Web, algorithmes de sélection, algorithme SPO, qualité de service, optimisation.

## Abstract

Due to the number and the enhancement of web services, the users have to claim many approaches which give them the possibility of a choice, even when the choice itself is a composed one. These approaches have to optimize many offers from different providers of QoS, more than that; it must satisfy general wishes of the user (e.g. global cost of compositions). In order to assure the aims, we deliver in this sheet work a hybrid meta-heuristic which combines particle swarms with evolving cross operators. The results seem to be satisfying and confirm the useful side of the meta-heuristic in this kind of problematic.

**Key word:** Web services, selection algorithms, SPO algorithm, quality of services, optimization.

## ملخص

عدد الخدمات على شبكة الإنترنت في تزايد ، ولذلك يحتاج المستخدمون إلى البحث عن النهج الذي يسمح لهم بتحقيق الاختيارات المطلوبة , خصوصا في حالة اختيار تركيبية معقدة من الخدمات . هذه المناهج تحتاج إلى تحسين سمات مختلفة من جودة الخدمة المقدمة من قبل الموردين , وأكثر يجب عليهم تلبية الاحتياجات الكلية للمستخدم (مثل أن التكلفة الإجمالية لتركيبية الخدمات). لضمان هذه الأهداف، فإننا نقترح في هذا العمل استدلال عالي هجين يجمع بين أسراب الجسيمات مع عمليات التقاطع المتطورة. نتائج مشجعة للغاية وتأكيد على جدوى الاستدلال العالي في هذا النوع من المشاكل.

**الكلمات الرئيسية:** خدمات الواب، خوارزمية الأسراب، الخوارزميات الوراثة، جودة الخدمة، التحسين

---

## Table de matières

Liste de figures .....	3
Introduction générale .....	5
Chapitre 1 : Les services Web .....	8
1. Introduction .....	9
2. L'architecture orientée service SOA .....	9
2.1. Définition.....	9
2.2. Les avantages de SOA.....	11
3. Les services web .....	11
3.1. Définition.....	11
3.2. L'intérêt des services Web .....	12
3.3. Les caractéristiques des services Web.....	13
3.4. L'architecture des services Web .....	13
3.5. Les principales technologies de développement de service Web .....	14
3.5.1. Communication .....	14
3.5.2. Description.....	18
3.5.3. Publication .....	19
3.6. Les avantages et inconvénients des services Web .....	20
3.6.1. Avantages.....	20
3.6.2. Inconvénients.....	21
4. Sélection de service web .....	21
5. Conclusion .....	23
Chapitre 2 : Conception et implémentation du prototype.....	24
1. Introduction .....	25
2. L'Algorithme PSO (Particle swarm optimization).....	25
2.1. Historique et définition .....	25
2.2. Principe .....	26
2.3. SPO et l'hybridation .....	28
2.4. Description de la base .....	30

---

<b>2.5. Description de la requête .....</b>	<b>31</b>
<b>2.6. La fonction objectif « Score ou fitness » .....</b>	<b>31</b>
<b>3. Conception .....</b>	<b>33</b>
<b>3.1. Diagramme de cas d'utilisation .....</b>	<b>33</b>
<b>3.2. Diagramme de séquence .....</b>	<b>34</b>
<b>3.3. Diagramme de classes .....</b>	<b>35</b>
<b>4. Interface Humain/Machine (IHM) .....</b>	<b>36</b>
<b>5. Expérimentation .....</b>	<b>40</b>
<b>6. Conclusion .....</b>	<b>42</b>
<b>Conclusion générale.....</b>	<b>43</b>
<b>ANNEXE .....</b>	<b>45</b>
<b>Références .....</b>	<b>47</b>

---

## Liste de figures

Figure 1. 1: architecture orientée services (SOA).....	11
Figure 1. 2: Architecture du service Web .....	14
Figure 1. 3 : Exemple Fichier XML de bibliothèque.....	16
Figure 1. 4 : Structure d'un message SOAP .....	17
Figure 1. 5 : Structure d'un document WSDL.....	18
Figure 1. 6 : les trois types de l'annuaire UDDI.....	19
Figure 1. 7 : Exemple de sélection (agence de réservation) .....	21
Figure 1. 8 : Exemple de motivation.....	22
Figure 1. 9 : Les approches de sélection .....	23
Figure 2. 1 : Schéma de principe du déplacement d'une particule .....	27
Figure 2. 2 : schéma de principe du déplacement d'une particule.....	27
Figure 2. 3 : L'organigramme de principe de SPO .....	28
Figure 2. 4 : croisement entre Gbest et X .....	29
Figure 2. 5 : L'organigramme de principe de SPO Hybride.....	30
Figure 2. 6 : Diagramme de cas d'utilisation.....	33
Figure 2. 7 : Diagramme de séquence.....	34
Figure 2. 8 : Diagramme de classes .....	35
Figure 2. 9 : fenêtre de login.....	36
Figure 2. 10 : message d'erreur de password.....	36
Figure 2. 11 : chargement de la base .....	37
Figure 2. 12 : validation des contraintes .....	37
Figure 2. 13 : message d'erreur de la base vide.....	38
Figure 2. 14 : sélectionner l'algorithme.....	38
Figure 2. 15 : l'affichage des résultats .....	39
Figure 2. 16 : l'affichage de temps d'exécution .....	39
Figure 2. 17 : histogramme de fitness et optimalité pour SPO classique .....	40
Figure 2. 18 : les fitness et leurs optimalité pour SPO .....	41
Figure 2. 19 : histogramme de fitness et optimalité pour SPO hybride.....	41
Figure 2. 20 : les fitness et leurs optimalité pour SPO hybride .....	42

---

## **Glossaire**

SW: service web.

SPO: Particle swarm optimization.

OEP : L'Optimisation par Essaim Particulaire

SOA: Service Oriented Architecture.

SOAP: Simple Object Access Protocol.

QOS : Qualité de service.

WSDL :Web Services Description Language.

UDDI: Universal Description, Discovery and Integration.

XML: eXtensible Markup Language.

W3C: World Wild Web Consortium.

HTTP: Hyper Text Transfert Protocol.

SMTP: Simple Mail Transfer Protocol.

FTP: File Transfer Protocol.

RPC : Remote Procedure call, un protocole d'appel de procédures à distance.

HTML: Hyper Text Markup Language.

REST: Representational State Transfer.

# Introduction générale

### **Contexte**

L'intérêt des services web est de permettre à une entreprise d'exporter à travers le réseau internet ses compétences et son savoir-faire, d'interagir avec ses partenaires, de rechercher de nouveaux marchés et de nouveaux supports de vente.

La constante progression d'utilisation des services Web vient du fait de leur fonctionnement sur les standards d'Internet. Ces services Web peuvent donc communiquer au travers des firewalls, déjà paramétrés pour internet, ne nécessitant pas de changement de règle de filtrage de sécurité. De plus ils sont prévus pour fonctionner au-dessus de l'Internet, ou la fiabilité et la vitesse ne peuvent pas être garantie. Ils sont également des composants qui ne se soucient pas de différentes plateforme sur lequel le système est réparti. Du fait de ce fonctionnement, Ils se moquent de l'absence de contrôle de déploiement de sorte que tous les demandeurs et fournisseurs soient promus immédiatement.

### **Problématique**

Plusieurs travaux ont été proposé dans la littérature, pour traiter les problèmes d'interaction et de composition des services web, en effet Le défi majeur dans ce domaine est que les services web sont faiblement couplées i.e., ces services sont développés par des entreprises qui sont totalement indépendantes les unes des autres.

Par conséquent les utilisateurs ont besoin d'approches qui permettent la sélection et la personnalisation de leurs compositions en utilisant les critères de QOS offerts par les fournisseurs de services.

Cette sélection doit optimiser au mieux les différents attributs de QOS, en plus elle doit prendre en compte les contraintes globales exigées par les utilisateurs

### **Contribution**

Dans le cadre de ce mémoire, nous proposons une approche qui se base sur les Méta-heuristiques mono-objectifs. Ces derniers sont répandus dans divers domaines pour résoudre des problèmes d'optimisation et de recherche. Nous nous inspirons de ces techniques pour décrire une approche de sélection multi-objective des meilleurs services Web.

## [Introduction générale]

---

Notre projet a pour le but d'assurer la sélection des services web a base de QOS (Qualité of service) en prenant une composition abstraite qui comporte n tâches et un ensemble de contraintes globales, le problème revient à instancié ces N taches avec des services concrets de tel sorte que les valeurs de QOS sont optimisées, et en même temps les contrainte globales de l'utilisateur sont satisfaites.

Pour arriver à cet objectif nous employons une version hybride des essaimes particulières (Particle swarm optimization), en effet notre contribution ajoute un opérateur de croisement sur les positions des particules, en plus des opérateurs standards, i.e. le mouvement individuel (basé sur la mémoire), et le mouvement social (i.e. l'information partagée par le groupe).

### **Plan du mémoire**

Le manuscrit est composé de deux chapitres et une conclusion générale, qui sont organisés comme suit:

Dans le **chapitre 1** nous avons présenté la technologie des services Web et les principaux standards qu'elle supporte, tout d'abord, nous commençons par l'architecture SOA (), ensuite, définir l'architecture des services web, après nous détaillons les différents protocoles d'un service web enfin nous citons quelque avantages inconvénients des services web afin de conclure ce que nous avons fait.

Le **chapitre 2** est consacré à la conception, l'implémentation et l'expérimentation du prototype. Tout d'abord nous présentons notre algorithme SPO, après nous expliquons une application à partir d'un scénario et mentionne les paramètres, les fonctions et la base de données utilisé, et enfin nous avons présenté la conception de notre système et l'implémentation pour évaluer la performance de l'approche proposée.

**La conclusion générale** résume les résultats de notre travail, et présente les perspectives que nous souhaitons réaliser dans le futur.

# Chapitre 1

## Les services Web

## **1.Introduction**

L'augmentation De nombreuses organisations tournent actuellement vers des architectures à base des services Web pour le développement et l'intégration d'applications ou des systèmes d'information. Les dernières décennies ont été marquées par le développement rapide des systèmes d'information distribués, et tout particulièrement par la diffusion de l'accès à Internet.

Cette évolution du monde informatique a entraîné le développement de nouveaux paradigmes d'interaction entre applications tels que la SOA. Cette dernière a été mise en avant afin de permettre des interactions entre applications distantes. L'architecture orientée service SOA est une méthode de conception basée sur des standards (SOAP, WSDL,...), permettant de créer une infrastructure informatique intégrée capable de répondre rapidement aux nouveaux besoins d'un utilisateur. Elle fournit les principes et directives permettant de transformer un réseau existant de ressources informatiques hétérogènes, distribuées, complexes et rigides en ressources intégrées, simplifiées et particulièrement souples pouvant être modifiées et combinées afin de mieux satisfaire les objectifs de l'utilisateur. [1]

Dans ce chapitre, nous présenterons l'architecture orientée services et sa principale réalisation : les services Web. Nous donnons les définitions des services Web et décrivons brièvement leurs principe standards et technologies. Enfin, nous abordons le concept de composition des services Web en explicitant les méthodes de sélection des services web.

## **2.L'architecture orientée service SOA**

### **2.1. Définition**

Plusieurs définitions sont utilisées pour définir et expliquer l'architecture SOA. Les définitions suivantes illustrent différentes vues de la SOA. Cependant, elles convergent toutes vers un seul sens :

➤ **Métier :**

*« L'architecture orientée service est un ensemble de méthodes techniques, métiers, procédurales, organisationnelles et gouvernementales pour réduire ou éliminer les frustrations avec les technologies d'information, et pour mesurer*

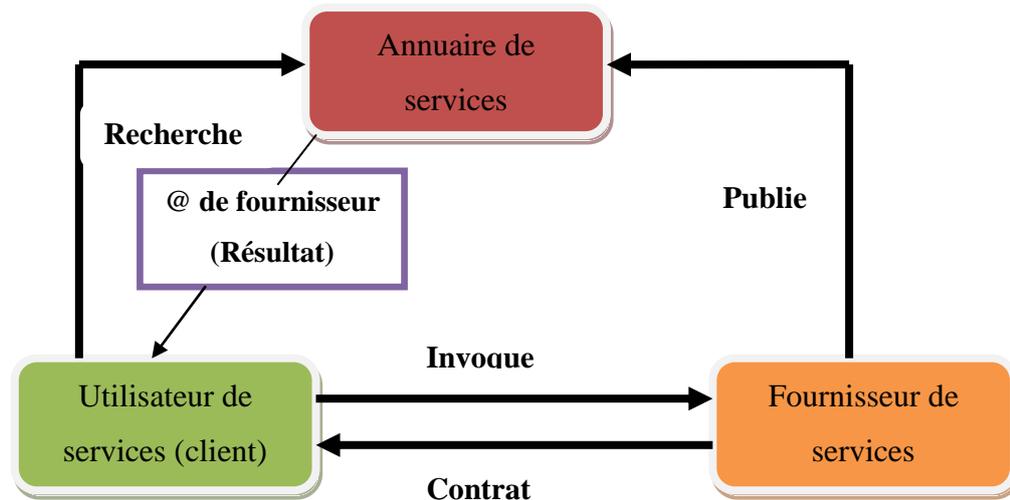
*quantitativement la valeur métier des technologies d'information, pendant la création d'un environnement métier agile pour un intérêt concurrentiel. » [2].*

➤ **Technique :**

*« Une architecture SOA est une structure d'intégration de processus métier qui supporte une infrastructure des technologies d'information comme étant des composants et services sécurisés, standardisés et qui peuvent être combinés pour s'adresser aux priorités de changements métiers. » [3].*

Par conséquent, un SOA (Services Oriented Architecture ; architecture orientée service) repose sur la réorganisation des applications à partir d'un ensemble de services élémentaires. Ces applications (visibles) représentent les services en question s'appuyant fonctionnellement parlant que des interfaces standards (langages SOAP/WSDL ou REST pour *Representational State Transfer*), connue sous le nom de Web Services, couches d'invocation compréhensibles potentiellement par l'ensemble des systèmes en présence, pour peu qu'elles intègrent le module d'interprétation nécessaire. Au sein d'un tel environnement, des services (dits "producteurs") sont ainsi exposés à d'autres services (dits "consommateurs"). [4]

Comme la montre la figure ci-dessous (**Figure 1.1**), les fournisseurs de services enregistrent et publient leurs services dans un annuaire de services pour respecter l'architecture SOA, les clients de services ou les utilisateurs consultent cet annuaire pour trouver des services qui vérifient les critères qui correspondent à certaines descriptions, si c'est vérifié l'annuaire répond aux clients en donnant les descriptions de la requête avec un contrat d'utilisation. Alors le client fait son choix en s'adressant au fournisseur pour invoquer le service Web. [5]



**Figure 1. 1: architecture orientée services (SOA)**

## 2.2. Les avantages de SOA

Une architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment :

- Une modularité permettant de remplacer facilement un composant (service) par un autre.
- Une réutilisabilité possible des composants (par opposition à un système tout-en-un fait sur mesure pour une organisation).
- De meilleures possibilités d'évolution (il suffit de faire évoluer un service ou d'ajouter un nouveau service).
- Une plus grande tolérance aux pannes.
- Une maintenance facilitée. [6]

## 3. Les services web

### 3.1. Définition

La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes. Les services Web sont basés sur le modèle SOA.

D'autres technologies telles que RMI (Remote method invocation), DCOM (Distributed Component Object Model) et CORBA ont auparavant adopté ce style architectural mais ont échoué en raison de la pluralité des plates-formes utilisées

dans les structures et aussi du fait que leur utilisation n'était pas adapté à Internet (problème de passage à travers des pare-feux, etc.) d'où le manque de souplesse, voire l'absence de réactivité sur ce réseau. Les applications réparties fondées sur ces technologies offrent des solutions caractérisées par un couplage « fort » entre les objets. Les solutions proposées par les services Web, permettent néanmoins un couplage « moins fort ». De plus, l'utilisation des technologies standards du Web telles HTTP et XML par les services Web facilite le développement d'applications réparties sur Internet, et permet d'avoir des applications très « faiblement » couplées. [7]

On peut citer plusieurs définitions :

- ❖ *D'après W3C (World Wide Web Consortium) : « Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet. »*
- ❖ *« Un service Web est une application accessible à partir du Web. Il utilise les protocoles Internet pour communiquer, et utilise un langage standard pour décrire son interface. » [8]*
- ❖ *« Les services Web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le Web. Les services Web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service Web est déployé, d'autres applications (y compris des services Web) peuvent le découvrir et l'invoquer. » [9]*

### 3.2. L'intérêt des services Web

Les services Web fournissent un lien entre applications. Ainsi, des applications utilisant des technologies différentes peuvent envoyer et recevoir des données au travers de protocoles compréhensibles par tout le monde. Les services Web sont **normalisés** car ils sollicitent les standards XML et généralement HTTP pour transférer des données et ils sont en affinité avec de nombreux autres environnements de développement. Ils sont donc exploitables à distance via

n'importe quel type de plate-forme. C'est dans ce contexte qu'un intérêt très spécial a été attribué à la conception des services Web puisqu'ils permettent aux entreprises de délivrer des applications profitables à distance par d'autres entreprises. [10]

### 3.3. Les caractéristiques des services Web

Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web qui possède les caractéristiques suivantes: [11]

- Il est accessible via le réseau.
- Il dispose d'une interface publique (ensemble d'opérations) décrite en XML ;
- **Web based** : les Web services sont basés sur les protocoles et les langages du Web, en particulier HTTP et XML.
- **Self-described, self-contained** : le cadre des Web services contient en lui-même toutes les Informations nécessaires à l'utilisation des applications, sous la forme de trois fonctions : trouver, Décrire et exécuter.
- **Modular** : les Web services fonctionnent de manière modulaire et non pas intégrée. Cela signifie qu'au lieu d'intégrer dans une seule application globale toutes les fonctionnalités, on crée (ou on récupère) plusieurs applications spécifiques qu'on fait inter-opérer entre elles, et qui remplissent chacune une de ces fonctionnalités. Une fonctionnalité développée sous forme de Web services peut dorénavant être réutilisée et recombinée à une suite d'autres fonctionnalités pour composer une Nouvelle application.
- Ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire.

### 3.4. L'architecture des services Web

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML), et les appliquent à des interactions entre machines. Jusqu'ici, l'accès via Internet à une ressource applicative ou à une base de données s'effectuait par l'envoi d'une requête s'appuyant sur des langages tels que (PHP, JSP, ...).

Comme pour le **World Wide Web**, les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune.

Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante.

Deux types d'architecture existent pour les services Web : La première dite architecture de référence, elle contient trois couches principales. La seconde architecture est plus complète, elle utilise les couches standards de la première architecture en ajoutant au-dessus d'autres couches plus spécifiques. Elle est appelée architecture étendue ou encore en Pile. [12]

La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment : [13]

- ✓ L'infrastructure de base (Discovery, Description, Exchange)
- ✓ Couches transversales (Security, Transactions, Administration, QoS)
- ✓ La couche Business Processus (BusinessProcess)

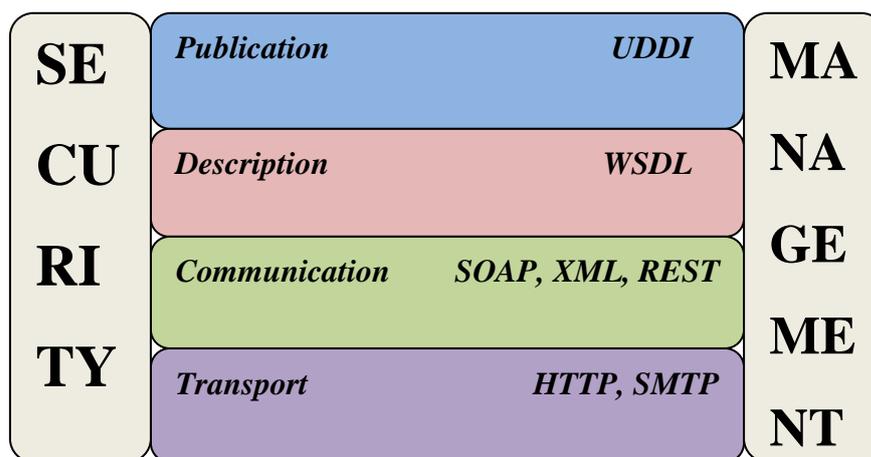


Figure 1. 2: Architecture du service Web

### 3.5. Les principales technologies de développement de service Web

Les technologies utilisées par les services Web sont XML, REST, SOAP, WSDL, UDDI et HTTP qui sont détaillé comme suit :

#### 3.5.1. Communication

##### ➤ XML

XML (*Extensible Markup Language*, ou Langage Extensible de Balisage) est un standard promulgué par le W3C, l'organisme chargé de standardiser les

## [Chapitre1 : les services web]

---

évolutions du Web et le langage destiné à succéder à HTML. Comme HTML (*Hypertext Markup Language*) c'est un langage de balisage (*markup*) : il représente de l'information encadrée par des balises. XML est un métalangage, ce qui veut dire que contrairement à HTML qui possède un ensemble de balises de présentation prédéfinies, il va permettre d'inventer de nouvelles balises d'isolement d'informations ou d'agrégats élémentaires que peut contenir une page Web. [14]

XML a été conçu pour des documents arbitrairement complexes, tout en s'appuyant sur cinq grands principes simples et clairs :

- Lisibilité à la fois par les machines et par les utilisateurs ;
- Définition sans ambiguïté du contenu d'un document ;
- Définition sans ambiguïté de la structure d'un document ;
- Séparation entre documents et relation entre document ;
- Séparation entre structure du document et présentation du document.

Le contenu d'un document XML est structuré par une suite d' «éléments», qui sont des blocs de texte encadré par des paires de balises ouvrantes et fermantes (<, >), qui sont les « unités de contenu ». Ces éléments sont liés entre eux par une hiérarchie, certains éléments apparaissant imbriqués dans d'autres (arborescence). XML et DTD permettent aussi la différenciation réelle du contenu de la structure de document. [15]

Voici un exemple de schéma XML définissant le type de document de la bibliographie. Ce schéma est volontairement rudimentaire pour un premier exemple. Il n'est pas très précis sur les contenus de certains éléments. Un exemple plus complet pourrait être donné pour la bibliographie.

Un schéma XML définit, d'une part, l'imbrication des éléments entre eux, ce qui s'apparente aux DTD, et d'autre part, le type des éléments et de leurs attributs. L'information fournie par le schéma est donc plus riche que celle dans le DTD. [16]

```
bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
    <publisher>Addison-Wesley</publisher>
    <price> 65.95</price>
  </book>
  <book year="1992">
    <title>Advanced Programming in the Unix
environment</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
    <abstract> </abstract>
  </book>
</bib>
```

**Figure 1. 3 : Exemple Fichier XML de bibliothèque**

### ➤ REST

REST (*Representational State Transfer*) est une architecture propre aux services Web. Élaborée en l'année 2000 par *Roy Fielding*, l'un des créateurs du protocole HTTP, du serveur Apache HTTP et d'autres travaux fondamentaux, REST est une alternative qui permet de bâtir une application pour les systèmes distribués comme le World Wide Web. [17]

### ➤ Le Protocol SOAP

SOAP (*simple Object Access Protocol*) est un protocole de communication défini à l'origine par Microsoft, puis standardisé par le W3C, avec l'élaboration de IBM. Un protocole pour l'échange d'information dans un environnement répartie, basé sur le standard *XML*.

SOAP permettant de définir les mécanismes d'échanges d'information entre des clients et des fournisseurs de services Web. Il s'appuie sur n'importe quel protocole de communication (HTTP, SMTP, FTP ...) pour transmettre les messages. [18]

SOAP définit un format pour l'envoi des messages. Les messages SOAP sont structurés en un document XML et deux parties obligatoires : l'enveloppe SOAP et le corps SOAP ; et une partie optionnelle : l'en-tête SOAP.

### ✓ Structure d'un message SOAP

Tout d'abord un message SOAP est un document XML qui doit avoir la forme suivante :

- **Envelope** (enveloppe) c'est lui qui contient le message et ses différents sous-blocs. Il s'agit du bloc racine XML. Il peut contenir un attribut *encodingStyle* dont la valeur est une URL vers un fichier de typage XML qui décrira les types applicables au message SOAP.
- **Header** (entête) c'est un bloc optionnel qui contient des informations d'entêtes sur le message. S'il est présent, ce bloc doit toujours se trouver avant le bloc Body à l'intérieur du bloc Envelope.
- **Body** (corps) c'est le bloc qui contient le corps du message. Il doit absolument être présent de manière unique dans chaque message et être contenu dans le bloc Envelope. SOAP ne définit pas comment est structuré le contenu de ce bloc. Cependant, il définit le bloc Fault qui peut s'y trouver.
- **Fault** (erreur) ce bloc est la seule structure définie par SOAP dans le bloc Body. Il sert à reporter des erreurs lors du traitement du message, ou lors de son transport. Il ne peut apparaître qu'une seule fois par message. Sa présence n'est pas obligatoire. [19]

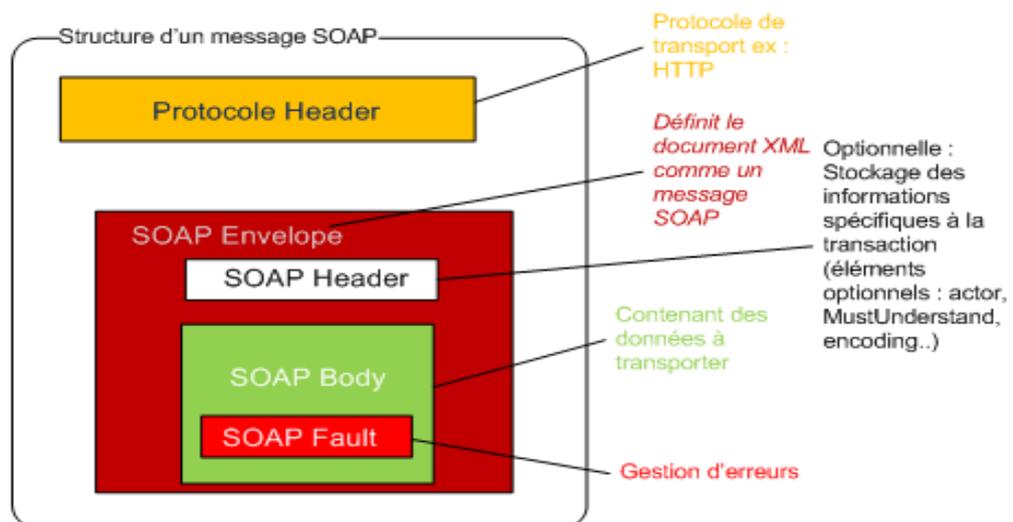


Figure 1. 4 : Structure d'un message SOAP

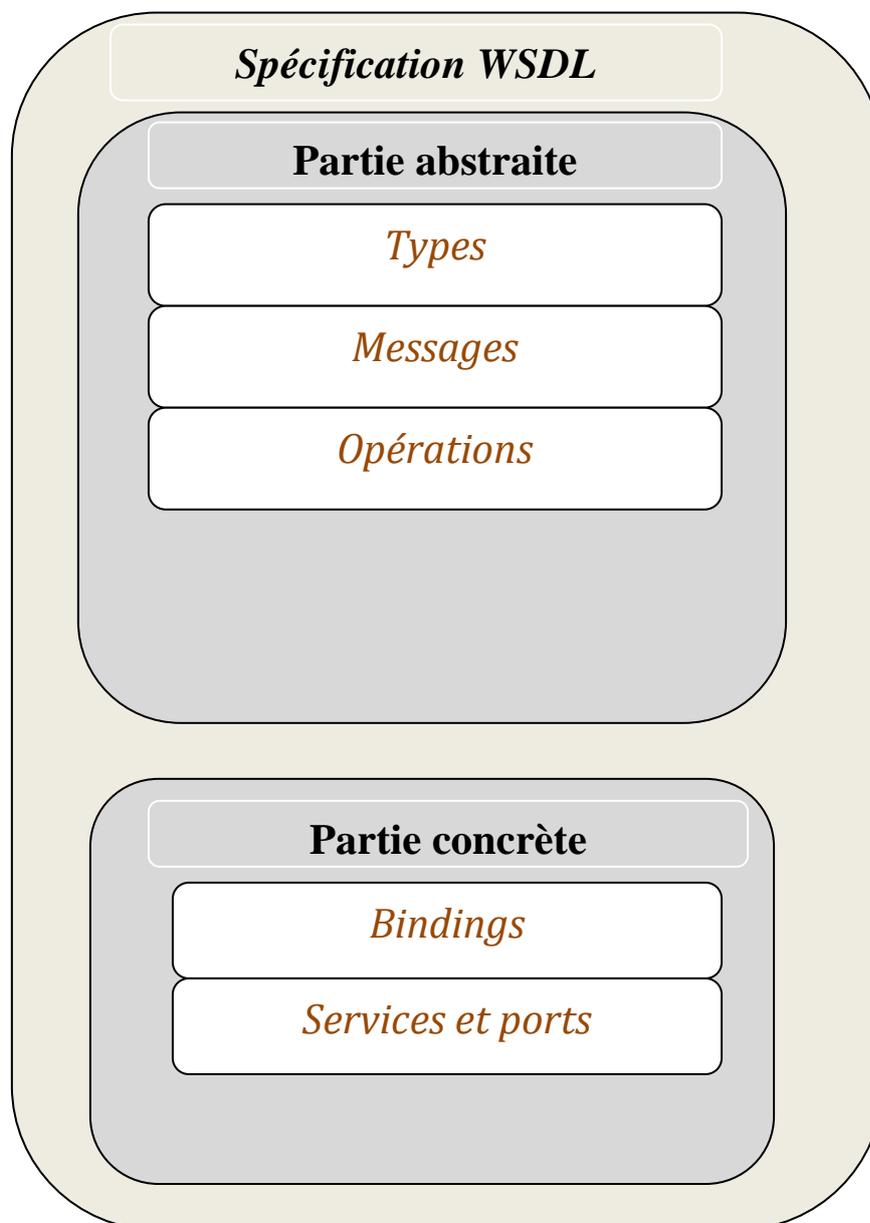
### 3.5.2. Description

➤ **La technologie WSDL (Web Service Description Language)**

✓ **Définition général**

WSDL est un langage qui permet de décrire les services web, et en particulier, les interfaces des services web. Ces descriptions sont documents XML. Le WSDL permettant de fournir les spécifications nécessaires à l'utilisation d'un service Web en décrivant les méthodes, les paramètres et ce qu'il retourne. [20]

✓ **Structure d'un document WSDL**



**Figure 1. 5 : Structure d'un document WSDL**

Définit la structure générale d'un document WSDL (**Figure 1.5**).

- **Types** : fournit la définition de types de données utilisés pour décrire les messages échangés.
- **Messages** : représente une définition abstraite (noms et types) des données en cours de transmission.
- **Opération** : c'est la description d'une action exposée dans le port.
- **Types de Port** : décrit un ensemble d'opérations. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou d'erreurs.
- **Binding** : spécifie une liaison entre un <portType> et un protocole concret (SOAP, HTTP...).
- **Service** : indique les adresses de port de chaque liaison.
- **Port** : représente un point d'accès de services défini par une adresse réseau et une liaison.

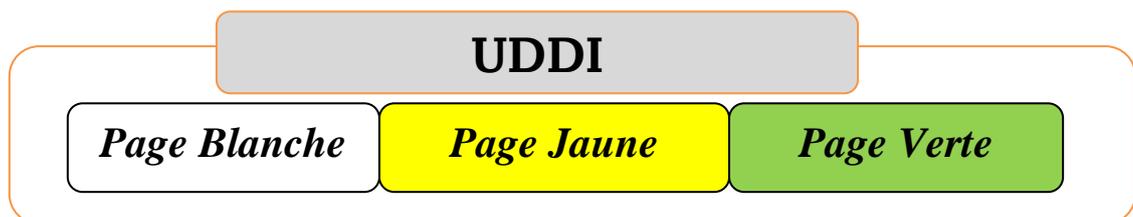
Le document WSDL peut être divisé en deux parties. Une partie pour les **définitions abstraites**, tandis que la deuxième contient les **descriptions concrètes**. [21]

### 3.5.3. Publication

#### ➤ UDDI (Universal Description Discovery and Integration)

##### ✓ Définition

UDDI (OASIS), a été conçu en 2000 à l'initiative d'un ensemble d'industriels (Ariba, IBM, Microsoft), en vue de devenir le registre standard de la technologie des services Web. Pour convenir à la technologie des services Web, les services référencés dans UDDI sont accessibles par l'intermédiaire du protocole de communication SOAP, et la publication des informations concernant les fournisseurs et les services doit être spécifiée en XML afin que la recherche et l'utilisation soient faites de manière dynamique et automatique. UDDI est une spécification définissant la manière de publier et de découvrir les services Web sur un réseau. [22]



**Figure 1. 6 : les trois types de l'annuaire UDDI**

Les informations qu'il contient peuvent être séparées en trois **types (Figure1.6)** :

- **Les pages blanches** incluant l'adresse, le contact et les identifiants relatifs aux services Web.
- **Les pages jaunes** identifiant les secteurs d'affaires relatifs aux services Web.
- **Les pages vertes** fournissent des informations techniques précises sur les services fournis.

Une fois ceci est fait, le service Web peut alors être connu de tous ceux qui le recherchent. Le modèle UDDI comporte quatre types de structures de données décrites sous forme de schéma XML : (**Business entity, Business service, Binding Tmodel, Publisher Assertion**). [23]

### 3.6. Les avantages et inconvénients des services Web

#### 3.6.1. Avantages

L'idée essentielle derrière les services Web est de partager les applications et les programmes en un ensemble d'éléments réutilisables appelés service, de sorte que, chacun de ces éléments effectuent une tâche principale et efficace, afin de faciliter l'interopérabilité entre tous ces services Web.

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Permettent de profiter de différents environnements et langages de développement par une publication, localisation, description et une invocation via XML. Les services
- Web sont très flexibles, indépendants des langages de programmation et des systèmes d'exploitation.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feux sans nécessiter des changements sur les règles de filtrage.
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants.

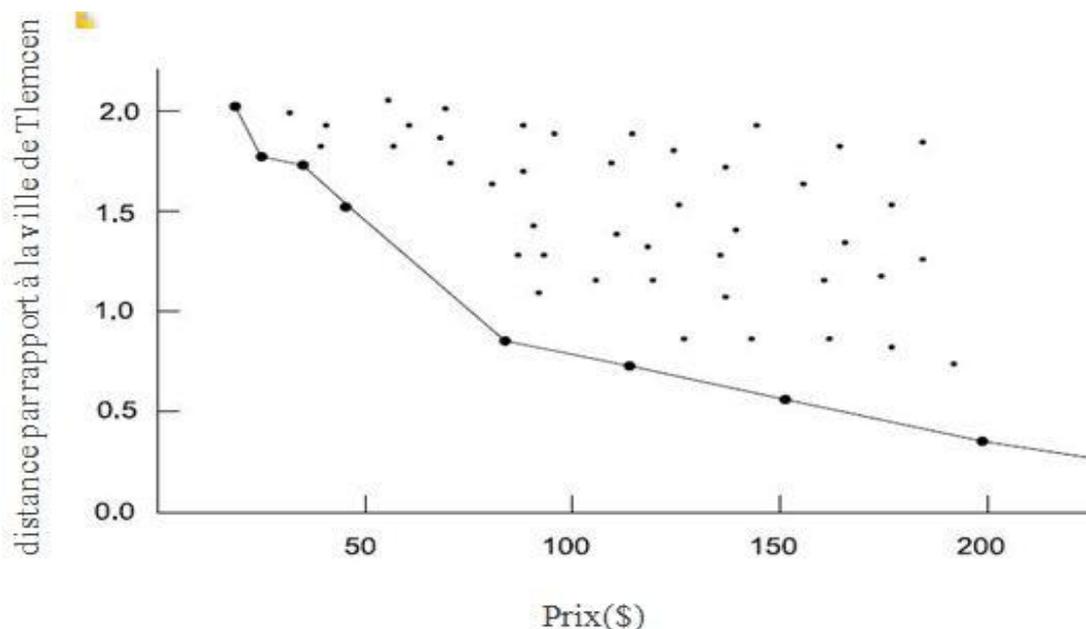
[24] [25]

### 3.6.2. Inconvénients

- La sémantique n'est pas prise en charge de façon efficace car le WSDL décrit les services de manière syntaxique.
- Les services Web ont de faibles performances par rapport à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.
- En l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place à travers les firewalls.
- Ils ne sont pas sécurisés à 100 %. [26]

### 4.Sélection de service web

Le domaine d'informatique a beaucoup évolué depuis ces années. L'émergence de l'approche à services a été accompagnée par une augmentation exponentielle du nombre de services et des fournisseurs. Avec la sélection des services web, on cherche à choisir le meilleur fournisseur d'un service web, étant donné un ensemble de fournisseurs de ce service. Par exemple une agence de réservation des hôtels, cette agence de voyage peut décider de composer plusieurs services Web afin de faciliter l'organisation d'un séjour. Le client veut un hôtel plus proche à la ville de Tlemcen avec un prix moins chère. On parle alors de composition de services Web. Il arrive très fréquemment que plusieurs services répondent à un même ensemble de besoins fonctionnels.



**Figure 1. 7 : Exemple de sélection (agence de réservation)**

Pour résumer nous considérons le problème de sélection de services web comme un problème d'optimisation combinatoire qui sera détaillé dans le chapitre suivant.

### ✓ Exemple de motivation

Nous présentons un déroulement dont l'objectif est d'illustrer et de motiver le problème de sélection des services web. Ce déroulement concerne la création d'une entreprise online pour faciliter le choix et aider le client à concrétiser les activités à travers un site web.

Pour cela nous avons besoin de 3 services :

- Un service qui fournit les informations nécessaires pour la création d'une entreprise.
- Un service qui fournit des informations sur les concurrences de marché de même type d'activité (exemple : TourismeAlgerie.com).
- Un service qui fournit des allocateurs pour l'hébergement de cette entreprise.

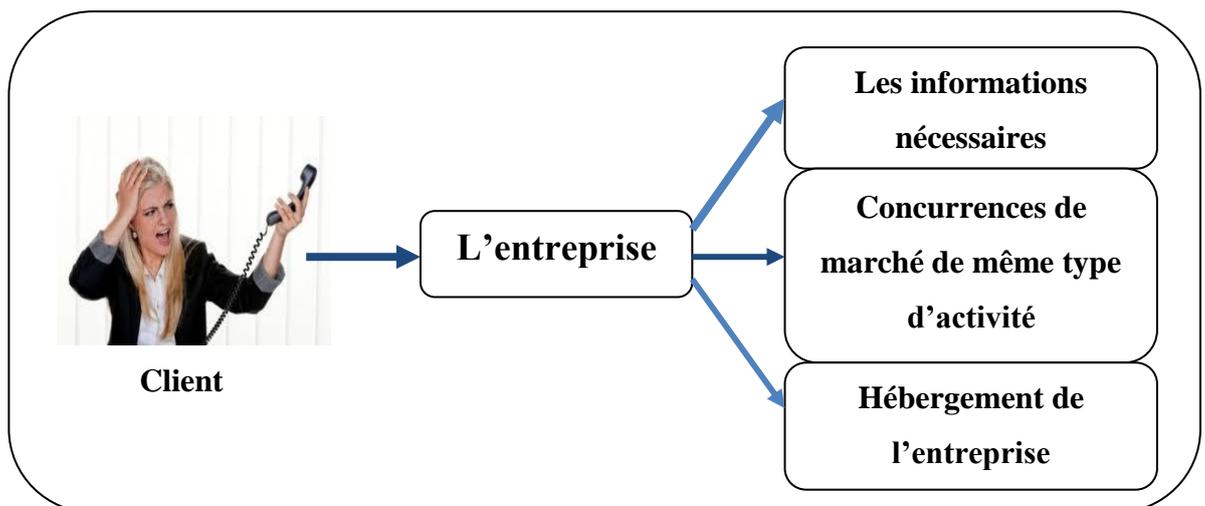
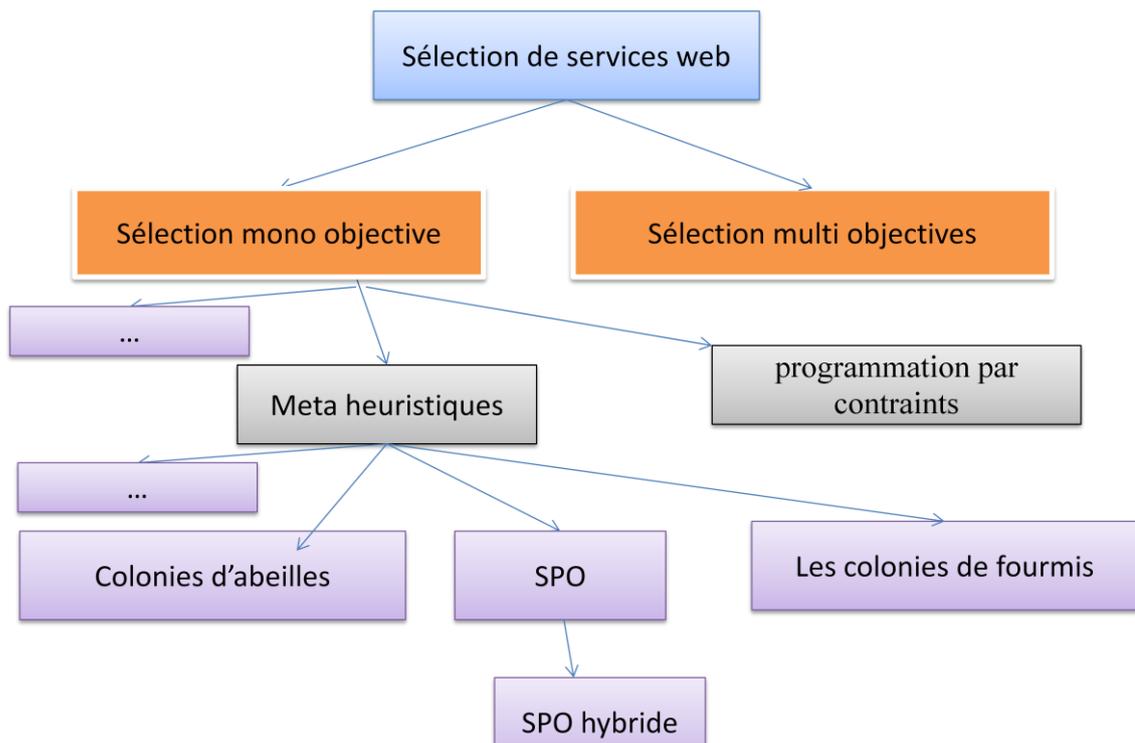


Figure 1. 8 : Exemple de motivation

### ✓ Les approches de sélection

La figure suivante montre l'état de l'art des différents travaux effectués sur le problème de sélection de service web.



**Figure 1. 9 : Les approches de sélection**

Plusieurs chercheurs ont créé des modèles interprétant le mouvement des vols d'oiseaux et des bancs de poissons. Plus particulièrement, Reynolds et Heppner et al. ont présenté des simulations sur un vol d'oiseaux. Reynolds était intrigué par l'aspect esthétique du déplacement des oiseaux en groupe et Heppner, un zoologue, était intéressé à comprendre les règles permettant à un grand nombre d'oiseaux de voler en groupe : soit de voler sans se heurter, de changer soudainement de direction, de s'écarter et de se rapprocher de nouveau. Cette étude a grandement inspiré le développement de l'algorithme PSO. [27]

## 5. Conclusion

Les services Web constituent une technologie idéale pour l'intégration et l'interopérabilité des systèmes répartis. Les services Web sont des applications accessibles par l'échange de documents XML entre deux URL. Ils sont caractérisés par leurs indépendances aux plates-formes et aux systèmes d'exploitation, ce qui a impliqué leur adoption par les différentes organisations commerciales et industrielles Offrant leurs services à travers le Web.

# Chapitre 2

Conception et  
implémentation du  
prototype

### **1. Introduction**

Avec l'augmentation et la prolifération des services web, la notion de qualité de service (QWS) émerge aujourd'hui et prend de plus en plus une grande importance pour les fournisseurs de service aussi bien que pour les clients. Notre application consiste à développer un système de sélection des services web à base de qualité de service.

L'approche des algorithmes évolutionnistes a fait l'effet d'une bombe dans les domaines de la résolution de problèmes complexes, et spécialement dans l'optimisation de fonction avec contraintes. Pour cela nous présentons une méta-heuristique apparue dernièrement : la méthode d'optimisation par l'essaim particulaire (OEP), en anglais : Particle swarm optimization (PSO).

PSO est une technique encore peu connue en monde, fondée sur la notion de coopération entre des agents (Les particules) qui peuvent être vus comme des « animaux » aux capacités assez limitées. L'échange d'information entre eux fait que, globalement, ils arrivent toutefois à résoudre des problèmes difficiles, comme c'est le cas, par exemple, chez les abeilles vivant en essaim (exploitation de sources de nourriture, construction de rayons, etc.).[27]

Dans ce dernier chapitre nous allons voir les différentes étapes suivies durant la réalisation de notre application, Nous commencerons d'abord par une représentation sur l'algorithme SPO (origine, principe...), puis nous présenterons une description de la base, après nous allons parler du processus de développement logiciel utilisé, en suit, nous finissons par présenter les différents diagrammes de conception ainsi que l'application et les résultats obtenus.

### **2. L'Algorithme PSO (Particle swarm optimization)**

#### **2.1. Historique et définition**

Les algorithmes de type Particle Swarm Optimization (PSO) sont des algorithmes d'optimisation probabilistes qui maintiennent au fil des itérations une population de solutions potentielles au problème. Ils ont été proposés par Eberhart et Kennedy en 1995 qui se sont inspirés du comportement collectif des vols d'oiseaux ou des bancs de poissons. En effet, lors de simulations des modèles mathématiques décrivant les vols d'oiseaux, Les algorithmes PSO partagent ainsi de nombreuses caractéristiques avec les algorithmes évolutionnaires comme les algorithmes génétiques [28].

## [Chapitre2: conception et implémentation des prototypes]

En utilisant les algorithmes de modélisation de Heppner [29] et de Reynolds [30], et en modifiant le modèle mathématique de Wilson, Kennedy et Eberhart ont transformé le tout en un vol d'oiseaux cherchant la « mangeoire » (suivre le meilleur voisin) la plus grosse dans un lot de mangeoires contenues dans une région prédéterminée. L'algorithme d'optimisation PSO a ainsi vu le jour. [31]

En effet, ils s'initialisent avec une population de solutions (prises aléatoirement) et cherchent un optimum au problème en faisant évoluer les individus de la population au fil des générations. A l'inverse des algorithmes génétiques où les solutions sont codées sous forme de chromosomes sur lesquels opèrent des opérateurs dédiés (mutation, crossover), les individus d'une population PSO sont directement les solutions au problème et la recherche des meilleures solutions se fait en déplaçant dans l'espace des solutions, ces individus ou particules. [32]

### 2.2. Principe

L'optimisation par essaim particulaire, Particle Swarm Optimization (PSO) en anglais, est une méta-heuristique basée sur une population de particules qui explorent l'espace d'un problème comme un groupe d'animaux à la recherche de nourriture. A partir des quelques informations dont elle dispose, une particule doit décider de son prochain mouvement, c'est-à-dire décider de sa nouvelle vitesse, comme on a montré dans **Figure 2.1** et **Figure2.2**. [30]

Pour ce faire, elle combine linéairement trois informations :

- Sa vitesse actuelle.  $V_i(t)$
- Sa meilleure performance.
- La meilleure performance de ses voisines (ses informatrices).

À l'aide de trois paramètres parfois appelés coefficients de confiance, qui pondèrent trois tendances :

- Tendance à suivre sa propre voie
- Tendance conservatrice (revenir sur ses pas)
- Tendance « panurgienne » (suivre le meilleur voisin). [31]

Donc les équations de base de l'algorithme SPO sont :

$$v_i(t+1) = v_i(t) + r_p(p_i - x_i) + r_g(p_g - x_i)$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

$x_i$  : la position courant

$r_p, r_g$  : random

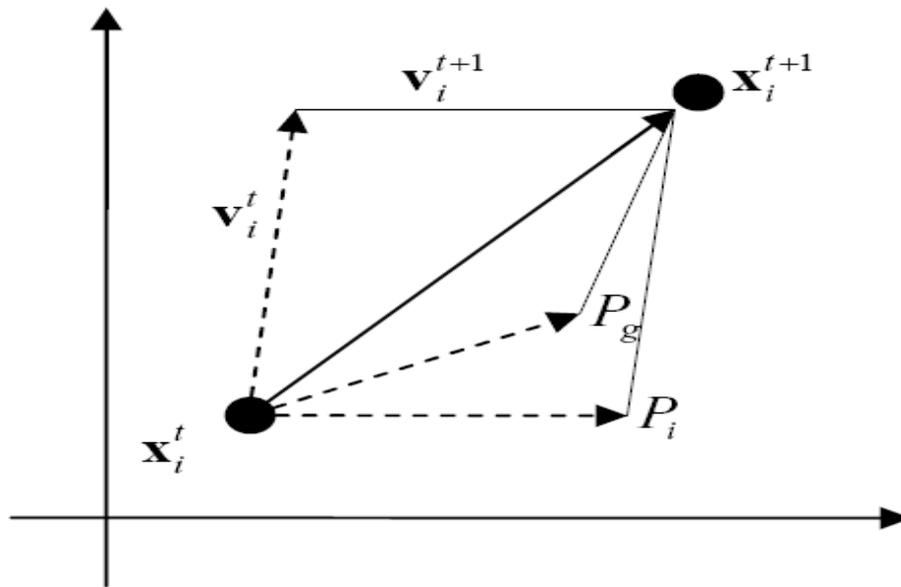


Figure 2. 1 : Schéma de principe du déplacement d'une particule

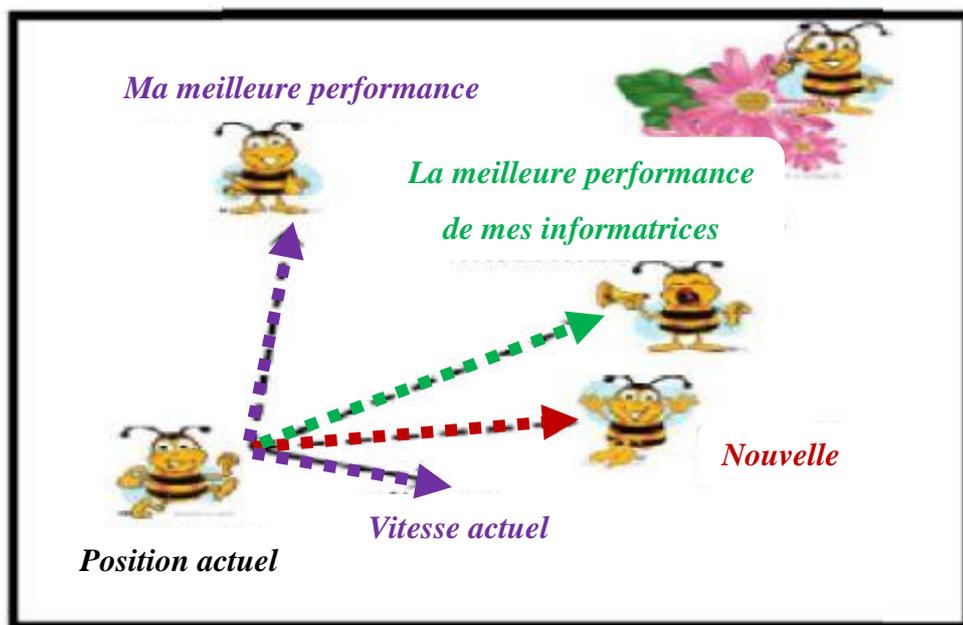
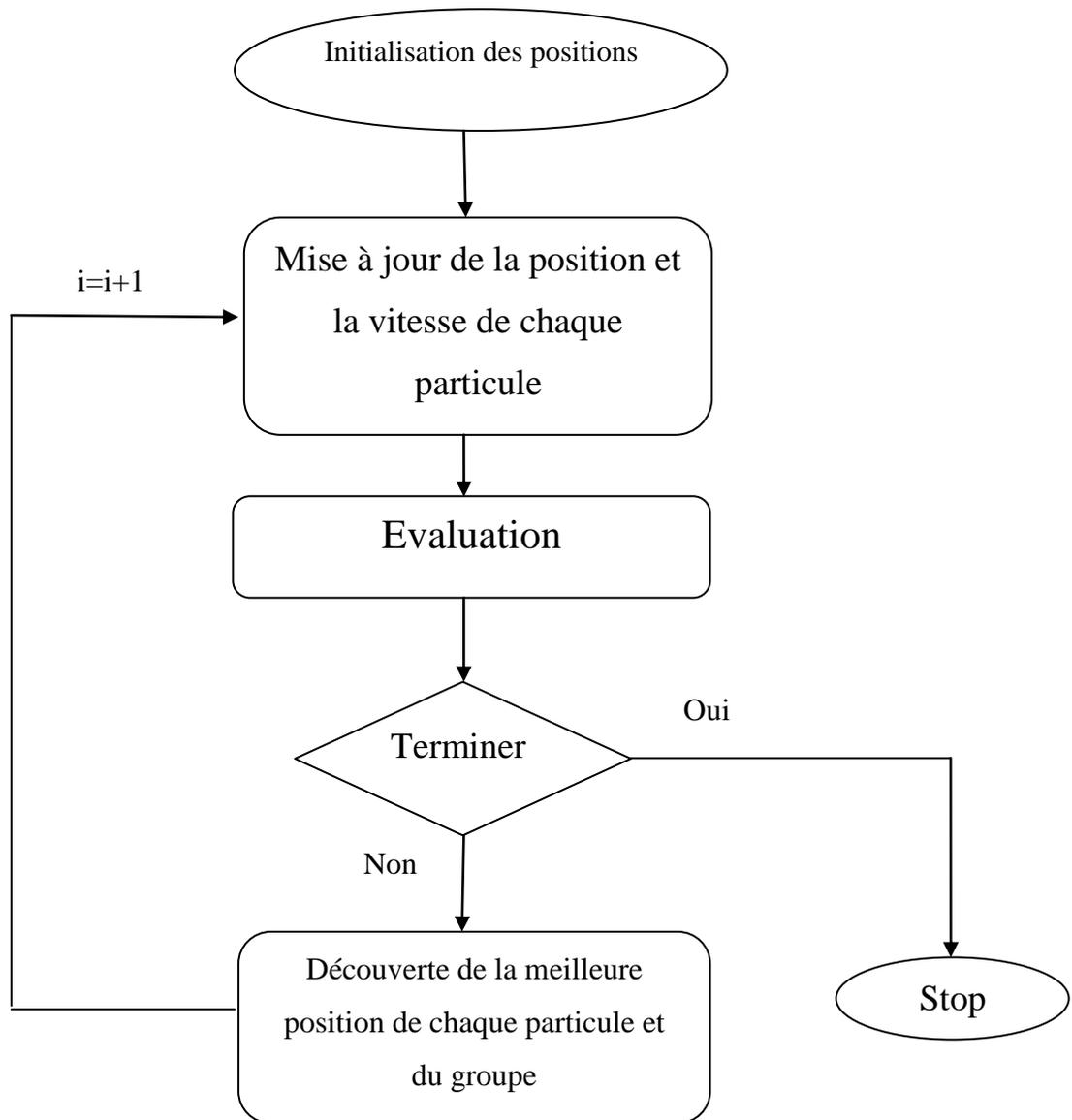


Figure 2. 2 : schéma de principe du déplacement d'une particule



**Figure 2. 3 : L'organigramme de principe de SPO**

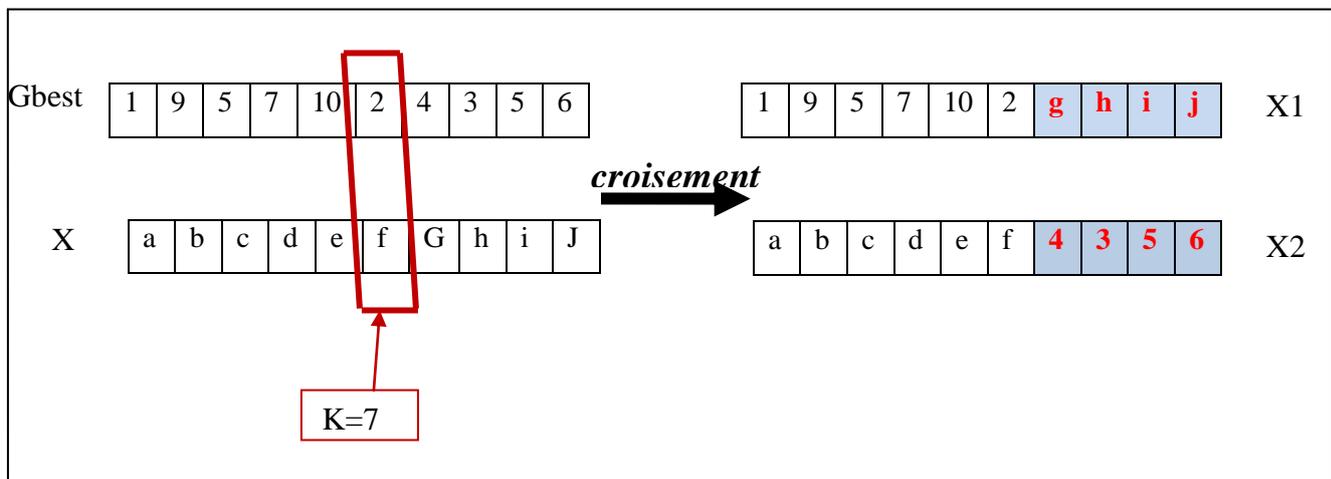
### 2.3. SPO et l'hybridation

Ces dernières années, l'hybridation des algorithmes a attiré l'attention de nombreux chercheurs afin d'améliorer leurs performances. L'objectif de l'hybridation est de combiner les caractéristiques de plusieurs algorithmes pour tirer profit de leurs avantages. Mais l'algorithme résultant risque d'hériter également de leurs faiblesses. De plus, un algorithme résultant de l'hybridation de plusieurs algorithmes peut avoir une complexité importante. Comme pour toutes les méta-heuristiques, l'hybridation a aussi touché le domaine de PSO dans le but d'améliorer ses performances. [32]

## [Chapitre2: conception et implémentation des prototypes]

Pour améliorer les performances de SPO traditionnel, nous ajoutons d'autres mouvements par rapport à ceux cités dans la section «2.2 principe». En particulier nous pouvons ajouter des opérateurs évolutionnaires tels que le croisement.

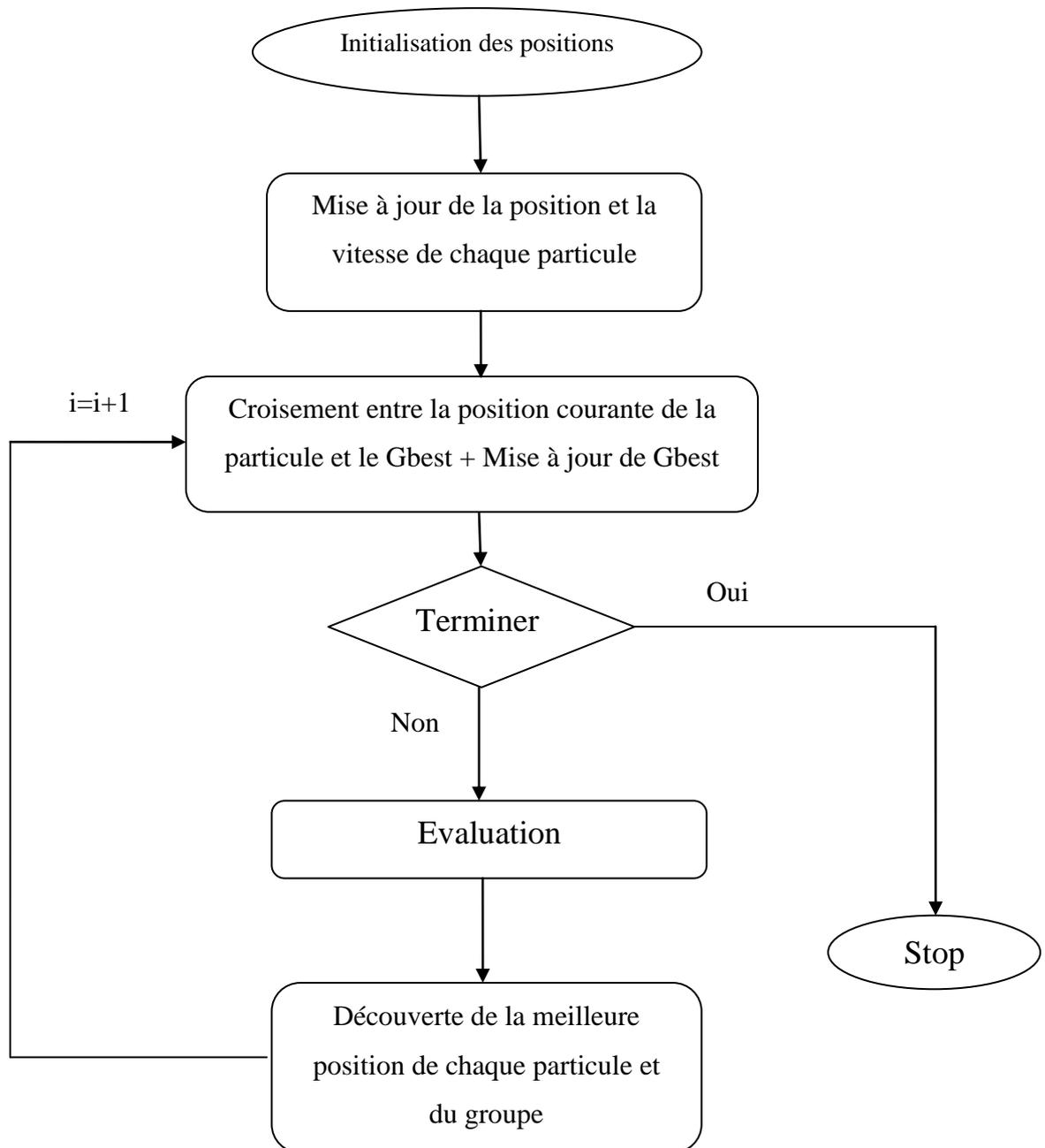
Dans notre travail, nous avons ajouté une étape de croisement entre la position courante  $X$  de la particule et la meilleure position de l'essaim  $G_{best}$ , un entier  $k$  représentant une position sur la chaîne est choisi aléatoirement entre 1 et la longueur de  $G_{best}$  (nombre de classe  $L$ ). Deux nouvelles enfants sont créées en échangeant tous les caractères compris entre les positions  $k + 1$  et  $L$ . L'exemple suivant **Figure 2.4** montre  $X$  et  $G_{best}$  de longueur  $L = \text{nombre de classe} = 10$  appartenant à la population initiale. Les deux nouvelles enfants ( $X1$  et  $X2$ ) appartenant à la nouvelle population sont obtenues par croisement à la position  $k = 7$  :



**Figure 2. 4 : croisement entre  $G_{best}$  et  $X$**

Les nouveaux enfants  $X1$  vont concurrencer l'ancien  $G_{best}$  pour devenir le nouveau  $G_{best}$ . Son rôle consiste à choisir aléatoirement deux individus parents parmi ceux sélectionnés pour les combiner et créer deux nouveaux individus enfants. Il joue un rôle important sur la convergence de l'AG, en lui permettant de concentrer une partie de la population autour des meilleurs individus.

Notre SPO hybride est illustré dans la **Figure 2.5** :



**Figure 2. 5 : L’organigramme de principe de SPO Hybride**

#### **2.4. Description de la base**

Nous avons créé un schéma de service qui contient dix (10) classes de services web. Le nombre d’instances dans chaque classe est quarante (40) fournisseurs de service web. Chaque fournisseur se caractérise par des qualités de service. Nous utilisons

## [Chapitre2: conception et implémentation des prototypes]

cinq (5) paramètres pour évaluer les opérations des services : délai, fiabilité, disponibilité, cout et réputation. La Latence(le délai) qui est une valeur aléatoire (entre 0 et 300 seconds), La Fiabilité qui est une valeur aléatoire entre (0.5 et 1), La Disponibilité qui est une valeur aléatoire entre (0.7 et 1), Le Coût qui est une valeur aléatoire entre (0 et 30 \$), et la Réputation qui est une valeur aléatoire (entre 0 et 5), et les lignes représentent les fournisseurs. Les valeurs de ces paramètres sont générés en se basant sur une distribution uniforme. Un échantillon de notre base est illustré dans la table suivant [33] qui représente le fournisseur  $i$  des cinq premières classes.

Attributs	L'intervalle
Cout	0 → 30 \$
Latence(Délai)	0 → 300 seconds
Disponibilité	0.7 → 1
Fiabilité	0.5 → 1
Réputation	0 → 5

### 2.5. Description de la requête

La requête de l'utilisateur comporte cinq (05) éléments :

1. La borne minimale la fiabilité. B1
2. La borne minimale la disponibilité. B2
3. La borne minimale la réputation. B3
4. La borne maximale pour le cout. B4
5. La borne maximale pour le délai (latence). B5

### 2.6. La fonction objectif « Score ou fitness »

La qualité d'un service web peut être définie comme un vecteur de QOS :

Qualité(S<sub>i</sub>) = (latence(S<sub>i</sub>), fiabilité(S<sub>i</sub>), disponibilité(S<sub>i</sub>), coût(S<sub>i</sub>), réputation(S<sub>i</sub>)).

$$Q_i(C) = \sum_{j=1}^n Q_i(S_j)$$

Le but de fonction de score c'est pour calculer une valeur hors du vecteur de QOS des services. Cela peut faciliter la comparaison de la qualité des

## [Chapitre2: conception et implémentation des prototypes]

services. Comme les utilisateurs peuvent avoir des préférences sur la façon dont leurs requêtes sont traitées. [33]

$F_{\text{ctobj}}(\text{sol}) = (\text{Cout}(\text{sol}), \text{Rep}(\text{sol}), \text{Disp}(\text{sol}), \text{Fiab}(\text{Sol}), \text{Temp}(\text{sol}))$

$$= \left( \sum_{Q_i \in \text{Neg}} W_i \frac{Q_i^{\text{max}} - Q_i}{Q_i^{\text{max}} - Q_i^{\text{min}}} + \sum_{Q_i \in \text{Pos}} W_i \frac{Q_i - Q_i^{\text{min}}}{Q_i^{\text{max}} - Q_i^{\text{min}}} \right) + \text{pénalité}(\text{sol})$$

$$\text{Où : } \text{Penalité}(\text{sol}) = -\sum_{i=1}^R (D_i^2)$$

Avec :

$$D_i = \begin{cases} 0 & \text{si } Q_i \geq B_i & \text{pour les critères négatifs} \\ |Q_i - B_i| & \text{sinon} \end{cases}$$

$$D_i = \begin{cases} 0 & \text{si } Q_i \leq B_i & \text{pour les critères positifs} \\ |Q_i - B_i| & \text{sinon} \end{cases}$$

Où  $Q_i$  est l'ième qualité de service,  $Q_i$  est calculé par la fonction d'agrégation, Neg et Pos sont les ensembles de QoWS négatifs et positifs respectivement. En négative (resp. positive) des paramètres, le plus élevé (resp. inférieure) de la valeur, le pire est la qualité.  $Q_i^{\text{max}}$  est la valeur maximale pour l'ième paramètre QoWS pour tous les plans possibles d'exécution du service et  $Q_i^{\text{min}}$  est le minimum.

### 3. Conception

#### 3.1. Diagramme de cas d'utilisation

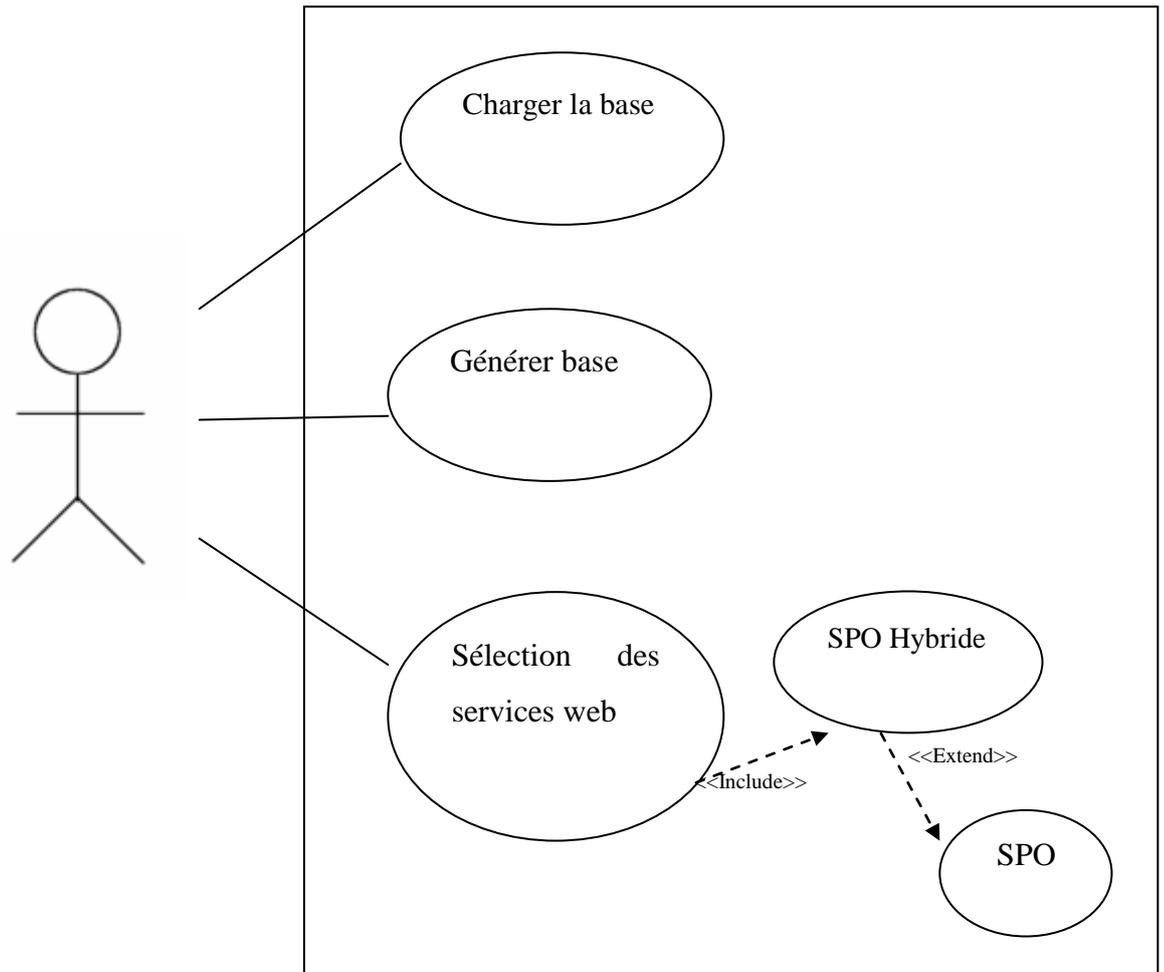


Figure 2. 6 : Diagramme de cas d'utilisation

### 3.2. Diagramme de séquence

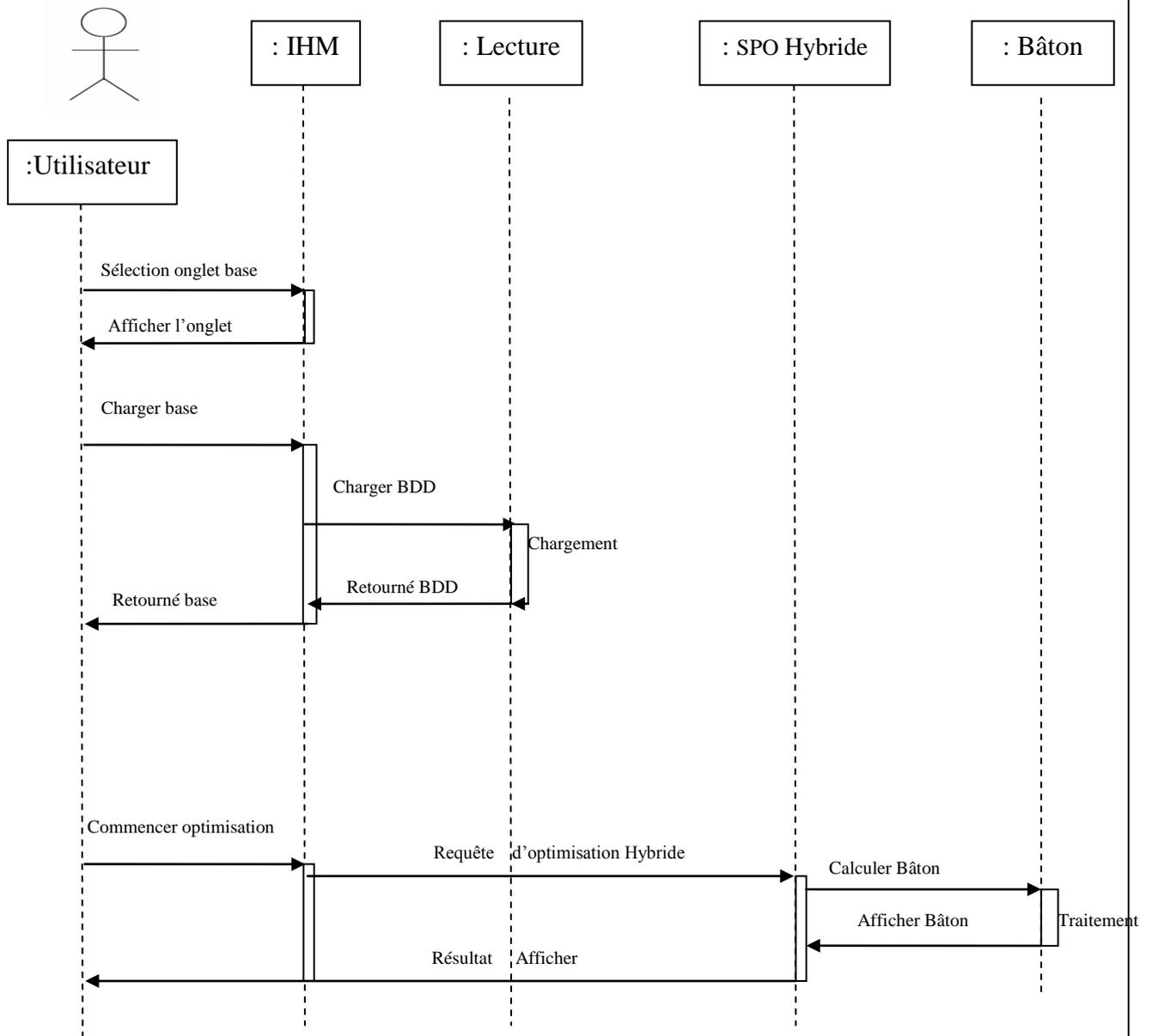


Figure 2. 7 : Diagramme de séquence

### 3.3. Diagramme de classes

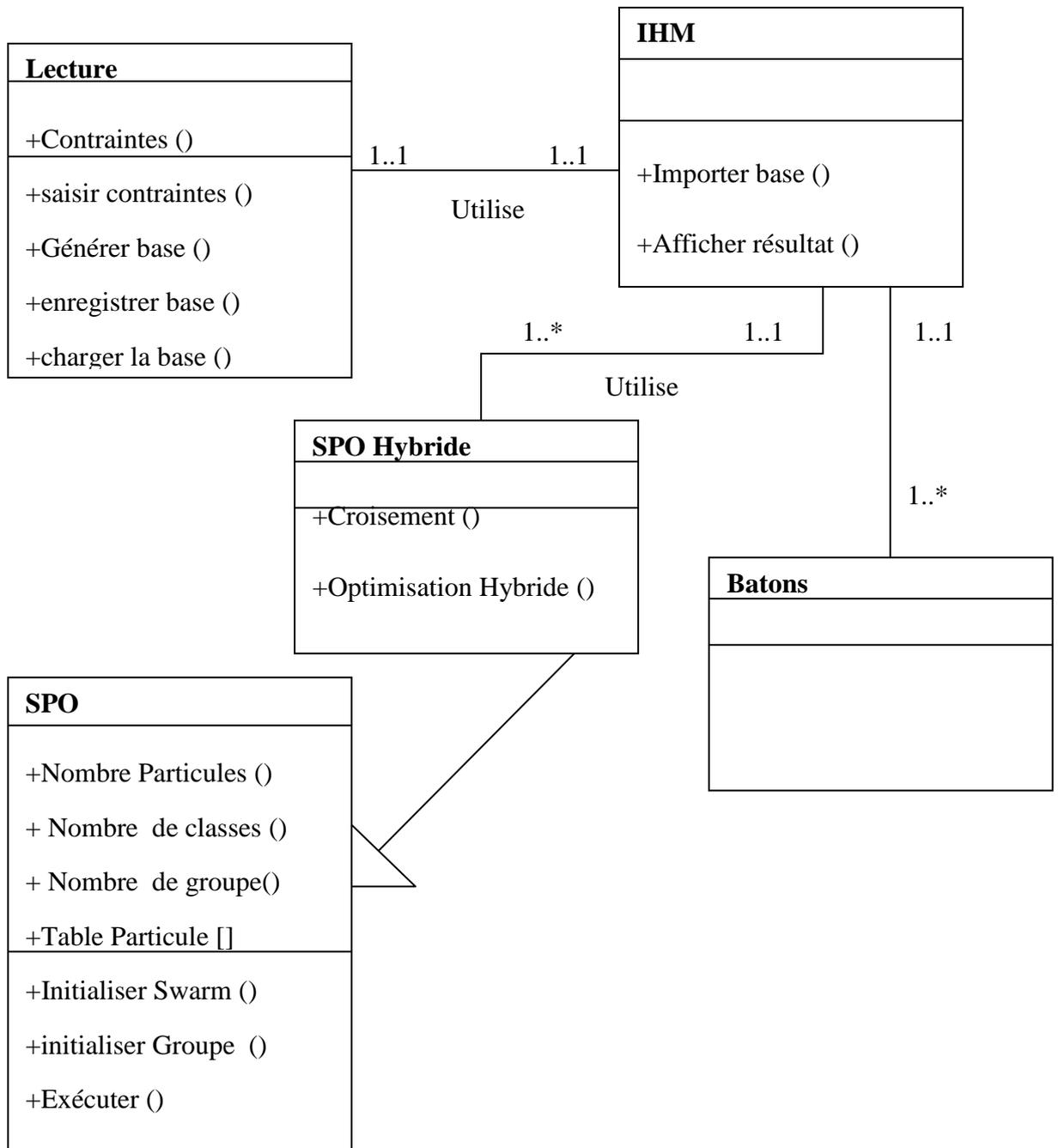


Figure 2. 8 : Diagramme de classes

#### 4. Interface Humain/Machine (IHM)

L'interface homme/machine représente l'élément clé dans l'utilisation de tout système informatique. Les interfaces de notre système de recherche sont conçues de manière à être simples, naturelles, compréhensible et d'utilisation faciles.

La première fenêtre qui s'exécute c'est la fenêtre de login comme on a montré dans la **Figure2.9** qu'elle demande un mot de passe pour accéder à l'application de l'algorithme (autorisation de l'administrateur). On a travaillé dans ce login avec Microsoft Access.

Si le mot de passe est incorrect il va afficher le message d'erreur (**Figure2.10**).



**Figure 2. 9 : fenêtre de login**



**Figure 2. 10 : message d'erreur de password**

La fenêtre principale se compose de cinq onglets principaux :

- Le premier onglet : pour le chargement, génération, enregistrement et affichage de la base de données

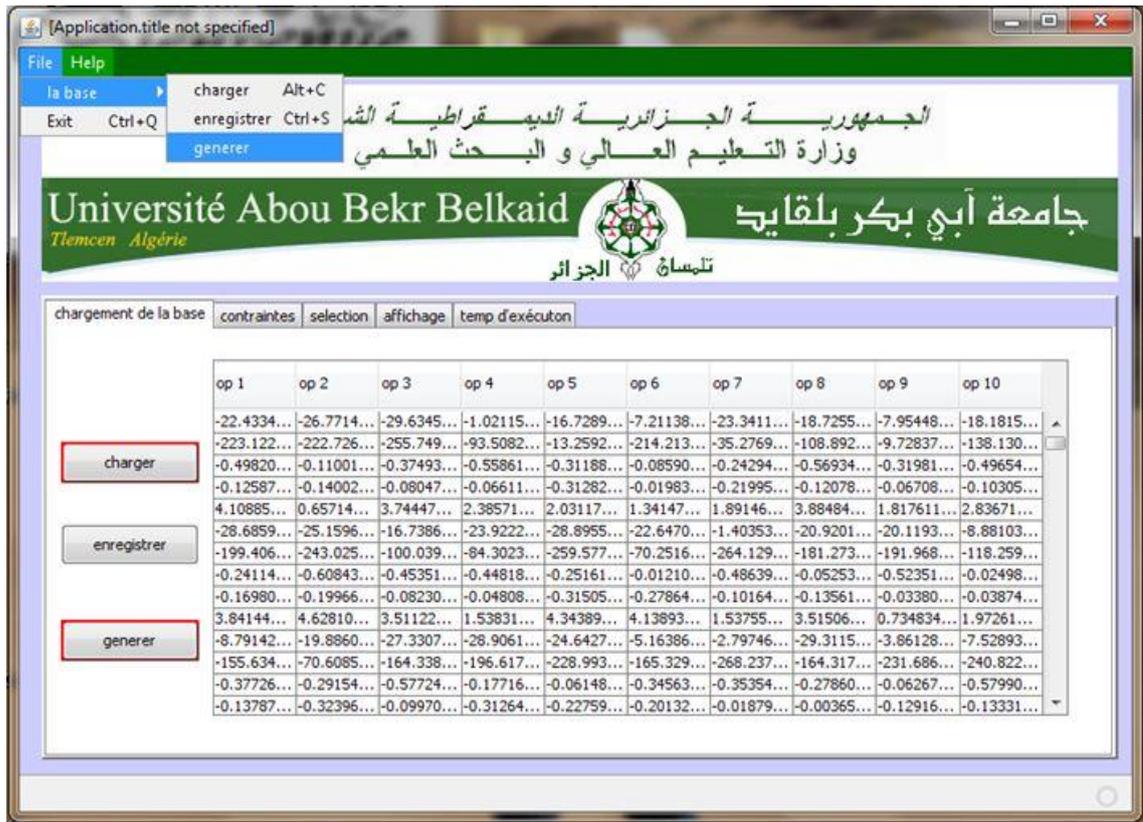


Figure 2. 11 : chargement de la base

- Le deuxième onglet : destinée pour la saisie et validation des contraintes.



Figure 2. 12 : validation des contraintes

## [Chapitre2: conception et implémentation des prototypes]

Le bouton « initialiser » doit initialiser automatiquement les valeurs des contraintes s'ils sont modifiés. Si l'utilisateur clique à le bouton de validation « ok» sans charger la base il va signaler le message d'erreur de « la base est vide» comme on a montré dans la **Figure 2.13**, même chose pour enregistrer et générer si la base est vide.



**Figure 2. 13 : message d'erreur de la base vide**

- Le troisième onglet : ce dernier onglet est destiné pour la sélection à base d'Essaim Particulaire, dont l'utilisateur a le choix de saisir le nombre d'itération et simulation qui lui convient. Après il doit choisir qu'elle algorithme d'essaim particulier (classique ou hybride) va exécuter pour la sélection.



**Figure 2. 14 : sélectionner l'algorithme**



Figure 2. 15 : l'affichage des résultats

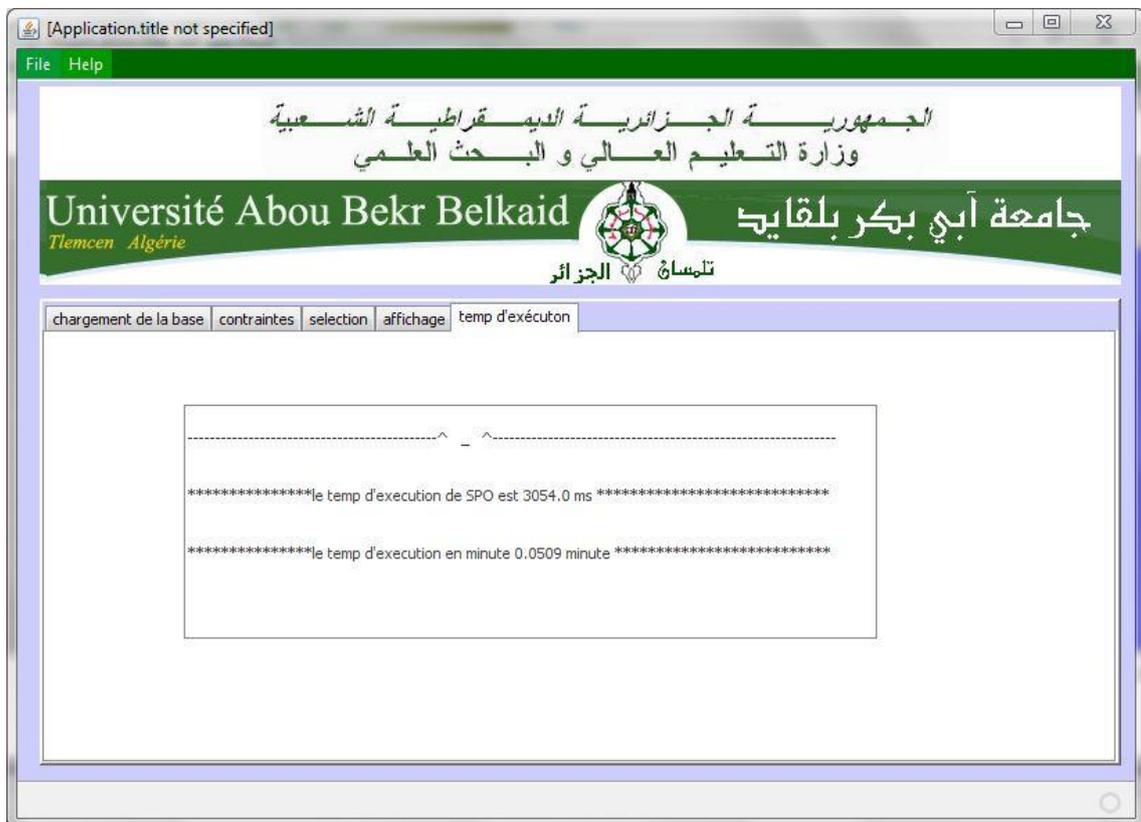
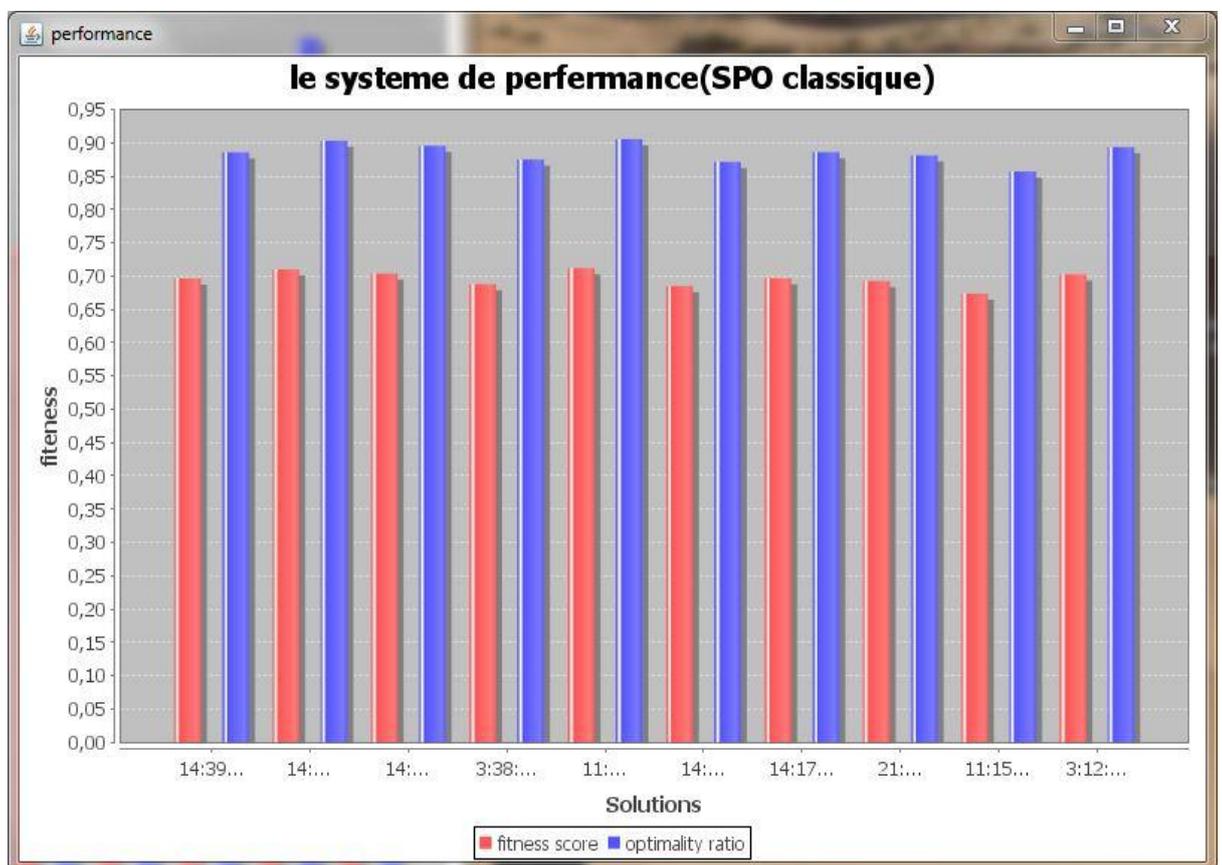


Figure 2. 16 : l'affichage de temps d'exécution

## 5. Expérimentation

Nous avons mené une expérience pour évaluer la performance de l'approche proposée. Notre but est de démontrer comment notre approche peut aider le client à sélectionner la meilleure offre. Nous courons notre expérience sous NetBeans IDE 7.0 de Sun Microsystems sous le système d'exploitation Windows, Processeur I3, 4 Giga de RAM.

**Figure 2. 17** montre 10 simulation de l'algorithme SPO avec 300 itérations, et **Figure2. 19** montres 10 simulations de l'algorithme SPO hybride avec 300 itérations.



**Figure 2. 17 : histogramme de fitness et optimalité pour SPO classique**

Optimalité
0.886
0.903
0.895
0.875
0.906
0.872
0.886
0.881
0.857
0.894

Figure 2. 18 : les fitness et leurs optimalité pour SPO

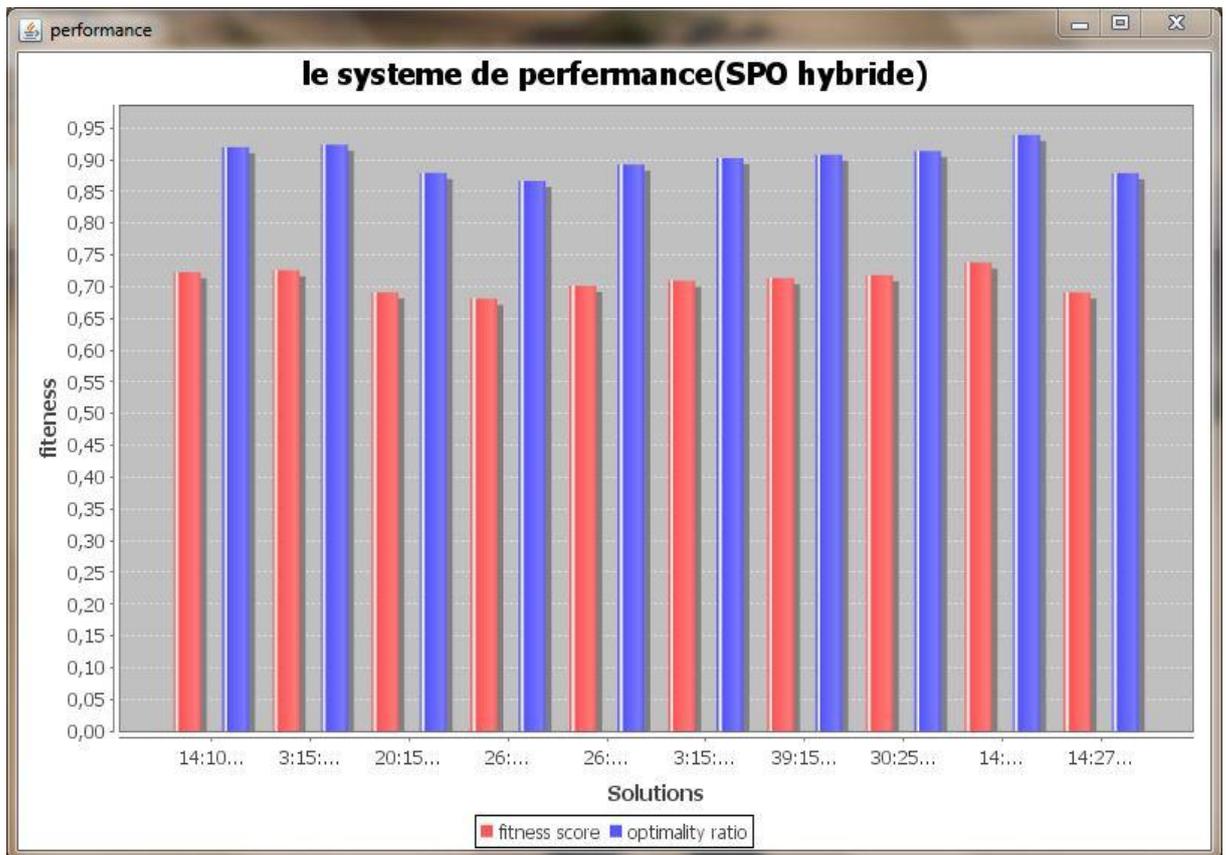


Figure 2. 19 : histogramme de fitness et optimalité pour SPO hybride

Optimalité
0.919
0.923
0.879
0.866
0.892
0.902
0.908
0.913
0.939
0.879

**Figure 2. 20 : les fitness et leurs optimalité pour SPO hybride**

➤ **Discussion**

On remarque que les résultats sont proche de l'optimale et on peut atteindre un taux plus que 90% pour les deux algorithmes SPO et SPO hybride.

En outre nous constatons que les pourcentages d'optimalité de SPO hybride sont plus grands que ceux de l'approche SPO classique. Par exemple on peut atteindre jusqu'à 93%.

## 6. Conclusion

Nous avons présenté dans ce chapitre deux algorithmes d'optimisation mono objectif à base des essaims particulière pour la sélection de services web. Nous avons proposé un exemple d'application pour illustré notre travail, et présenté les résultats de la validation de notre approche, afin d'évaluer l'efficacité de cette dernière.

D'après les résultats qu'on a obtenus, on a confirmé l'efficacité des algorithmes SPO hybride.

# Conclusion générale

## [Conclusion générale]

---

Nous avons présenté dans ce mémoire les technologies liées aux services Web. Nous avons proposé aussi un algorithme de sélection qui se base sur d'Optimisation par Essaim Particulaire hybride. L'optimisation par essaim particulaire est une méta-heuristique destinée à la résolution de problèmes à variables continues, inspirée du comportement en essaim de certains animaux.

Notre prototype sélectionne les compositions de services les plus satisfaisantes, en se basant sur cinq(05) critères de QOS : Latence, fiabilité, disponibilité, coût, réputation. La composition concrète recherchée doit maximiser un ensemble de ces critères positifs et minimiser un ensemble d'autres critères négatifs, en plus elle doit satisfaire un groupe de contraintes globales. Ensuite on a hybridé notre algorithme avec le croisement de quelques particuliers.

Tout travail est amené à être amélioré, en ce sens, notre système peut encore évoluer et se voir améliorer. Une voie de recherche pour l'amélioration de cette méthode est d'appliquer d'autres algorithmes d'optimisations on peut les prendre comme perspective à ce travail :

- ✓ Les colonies de fourmis.
- ✓ Colonies d'abeilles.
- ✓ La programmation par contraintes.
- ✓ .....

# **ANNEXE**

✓ **Outils et environnement de développement**

Nous allons spécifier les outils utilisés qui nous ont semblé être un bon choix vu les avantages qu'ils offrent.

➤ **Langage JAVA**

Pour le langage de programmation notre choix s'est porté sur le langage JAVA, et cela parce que JAVA est un langage orienté objet simple ce qui réduit les risques d'incohérence; il est portable, il peut être utilisé sous Windows, sous Linux, sous Macintosh et sur d'autres plateformes sans aucune modification. Java utilise des processus qui augmentent les performances des entrées/sorties, facilitent l'internationalisation. Il examine le programme au fil de l'exécution et libère automatiquement la mémoire, enfin il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards.

➤ **NetBeans**

Nous avons utilisé un éditeur de Java appelé NetBeans version 6.8 sous Windows qui est placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactorisations, éditeur graphique d'interfaces et de pages Web).

✓ **W3C (World Wide Web Consortium)**

Consortium industriel visant à favoriser l'interopérabilité des produits informatique et l'évolution du World Wide Web, par le développement de standards et de spécifications techniques.[33]

✓ **Protocoles**

Ensemble de règles qui sont partagée entre deux ordinateurs afin de communiquer l'un avec l'autre.

➤ **HTTP** (Hypertext Transfer Protocol) : Protocole de communication permettant l'échange de fichiers sur le World Wide Web. HTTP est un protocole d'application développé par le W3C.

➤ **SMTP** (Simple Mail Transfer Protocol (littéralement) : est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique.

### Références

- [1] Adel Boukhadra, La composition dynamique des services Web sémantiques a base d'alignement des ontologies owl-s, 2011
- [2] Ben Margolisand, et Joseph Sharpe, SOA for the Business developer: Concepts, BPEL, and SCA, édition MC Press, October 2007.
- [3] Frank Jennings, Ramesh Loganathan, et Poornachandra Sarang, Approach to Integration; XML, Web services, ESB, and BPEL in real-world SOA projects, édition Packt Publishing Ltd, November 2007.
- [4] Antoine Crochet-Damais "SOA pour Service Oriented Architecture : décryptage" , 2007
- [5] William Brown, Robert G. Laird, Clive Gee, Tilak Mitra : SOA Governance Achieving and Sustaining Business and IT Agility 2008
- [6] <http://www.commentcamarche.net/contents/web-services/soa-architecture-orientee-services.php3> Juin 2013.
- [7] <http://www.siteduzero.com/informatique/tutoriels/les-services-web/qu-est-ce-qu-un-service-web> 8 janvier 2013.
- [8] Tarek Melliti, Interopérabilité des Services Web complexes, Thèse de Doctorat, Université Paris IX Dauphine, le 8 Décembre 2004.
- [9] Julien Ponge. Model Based Analysis of Time-aware Web Services Interactions. Thèse de Doctorat de l'Université Blaise Pascal - Clermont-Ferrand II, dans le cadre de l'Ecole Doctorale des Sciences pour l'Ingénieur, France, 2008.
- [10] Heather Kreger, Web Service Conceptual Architecture, édition IBM Software Group, Mai 2001.
- [11] Ethan Cerami, Web Services Essentials, édition O'Reilly, Février 2002.
- [12] Heather Kreger, Web Service Conceptual Architecture, édition IBM Software Group, Mai 2001.
- [13] Heather Kreger, Web Service Conceptual Architecture, édition IBM Software Group, Mai 2001.

## [Références]

---

- [14] <http://www.ecr-france.org/boite-a-outils/glossaire-ecr/341-xml-extensible-markup-language-?lang=> 2011.
- [15] Chauvet J., “Service Web avec SOAP, WSDL, UDDI, ebXML“; jouve, Paris; mars 2002 .
- [16] SoftDeath : Architecture d'un service Web 2003.
- [17] <http://www.siteduzero.com/informatique/tutoriels/les-services-web/architecture-d-un-service-web> 8 janvier 2013.
- [18] Batiste Bieler, Etude de faisabilité d'une application SOAP avec un système embarqué, école HE-ARC ingénierie informatique, 2005.
- [19] W3C World Wide Web Consortium; ‘SOAP Version 1.2 Part 1: Messaging Framework’; Disponible à : <http://www.w3.org/TR/soap12-part1/> 27 avril 2007
- [20] Roberto Chinnici, et Martin Gudgin, Web Services Description Language, édition W3C, le 22 Aout 2004.
- [21] Chinnici R., Gudgin M., Moreau J. and Weerawarana S. “Web Services Description Language (WSDL) Version 1.2 ”, W3C Working Draft January 2003,
- [22] Site officiel d’UDDI: <http://www.uddi.org/> 2006-09-18.
- [23] Céline Lopez-Velasco, Sélection et composition de services Web pour la génération d’applications adaptées au contexte d’utilisation, pour obtenir le grade de Docteur de l’univesité JOSEPH FOURIER, 18 novembre 2008.
- [24] [http://fr.wikipedia.org/wiki/Service\\_Web#Avantages](http://fr.wikipedia.org/wiki/Service_Web#Avantages) 14 juin 2013.
- [25] David Chappell, et Tyler JEWELL, Java Web Service, édition O’Reilly, Mars 2002.
- [26] <http://www.w3.org/2001/XMLSchema-instance> 2001/03/16.
- [27] Kennedy J. , Eberhart R.C, Particle Swarm Optimization, Proceedings of the IEEE International Conference On Neural Networks, pages 1942-1948, IEEE Press, 1995.
- [27] Zeyneb KAMECHE : Sélection des Web Services à Base Des Essaimes Particulaires du Master en Informatique, Modèle intelligent et décision 2011.

## [Références]

---

- [29] Clerc M. , TRIBES - Un exemple d'optimisation par essaim particulaire sans paramètres de contrôle, Conférence OEP'03, Paris, France, 2 Octobre, 2003.
- [30] Heppner, F. And Grenander, U., A stochastic nonlinear model for coordinated bird flocks. In S. Krasner, Ed., the Ubiquity of Chaos. AAAS Publications, Washington, DC, pages 233-238, 1990.
- [31] Reynolds, C.W. , Flocks, herbs and schools : a distributed behavioral model, Computer Graphics, pages 25-34, 1987.
- [32] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm intelligence, San Francisco, Morgan Kaufmann, 2001.
- [33] Othmane bouguettaya : Foundation dor Efficient web Service Selection 2009
- [34] Hamane Bekhada: L'application de la méthode d'essaim particulaire 2008-2009.
- [35] Maurice Clerc : L'optimisation par essaim particulaire, Tutoriel pour OEP 2003.
- [36] Abbas EL DOR: Perfectionnement des algorithmes d'Optimisation par Essaim Particulaire. Applications en segmentation d'images et en électronique 2012.
- [37] W3C World Wide Web Consortium; : <http://www.w3.org/> .

## Résumé

Le nombre de services web est en augmentation constante sur le réseau internet, par conséquent, les utilisateurs ont besoin d'approches qui permettent leur sélection, surtout dans le cas des compositions de services. Ces approches doivent optimiser les différents attributs de QOS offerts par les fournisseurs, en plus elles doivent satisfaire les exigences globales de l'utilisateur (tels que le cout total des compositions).

Pour atteindre ces objectifs, nous proposons dans ce travail une méta-heuristique hybride qui combine les essaimes particulières avec les opérateurs de croisement évolutionnaire. Les résultats obtenus sont très encourageant et confirment l'utilité des méta-heuristiques dans ce genre de problèmes.

**Mots clés :** services Web, algorithmes de sélection, algorithme SPO, qualité de service, optimisation

## Abstract

Due to the number and the enhancement of web services, the users have to claim many approaches which give them the possibility of a choice, even when the choice itself is a composed one. These approaches have to optimize many offers from different providers of QoS, more than that; it must satisfy general wishes of the user (e.g. global cost of compositions). In order to assure the aims, we deliver in this sheet work a hybrid meta-heuristic which combines particle swarms with evolving cross operators. The results seem to be satisfying and confirm the useful side of the meta-heuristic in this kind of problematic.

**Keywords:** Web service, selection algorithms, SPO algorithm, quality of service, optimization.

## ملخص

عدد الخدمات على شبكة الإنترنت في تزايد ، ولذلك يحتاج المستخدمون إلى البحث عن النهج الذي يسمح لهم بتحقيق الاختيارات المطلوبة , خصوصا في حالة اختيار تركيبية معقدة من الخدمات .

هذه المناهج تحتاج إلى تحسين سمات مختلفة من جودة الخدمة المقدمة من قبل الموردين , وأكثر يجب عليهم تلبية الاحتياجات الكلية للمستخدم (مثل أن التكلفة الإجمالية لتركيبية الخدمات). لضمان هذه الأهداف، فإننا نقترح في هذا العمل استدلال عالي هجين يجمع بين أسراب الجسيمات مع عمليات التقاطع المتطورة. نتائج مشجعة للغاية وتأكيد على جدوى الاستدلال العالي في هذا النوع من المشاكل.

**الكلمات الرئيسية:** خدمات الواب، خوارزمية الأسراب، الخوارزميات الوراثة، جودة الخدمة، التحسين