

République Algérienne Démocratique et Populaire  
Université Abou Bekr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

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

*Thème*

## Sélection de services web par l'Algorithme Tabou

Réalisé par :

- Boufatah Abdesselam
- Belaidi Soufiane

*Présenté le 29 Juin 2013 devant le jury composé de MM.*

- *MR.SMAHI Mohammed Ismail* (Président)
- *Mr. HADJILA Fethallah* (Examineur)
- *Mr. BELABED Amine* (Examineur)
- *Mr. MERZOUG Mohammed* (Encadreur)

## *Remerciements*

*Nous tenons à remercier :*

*Allah le tous puissant.*

*Mr. MERZOUG Mohammed, 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. SMAHI.M.I*
- ❖ *Mr .HADJILA FATHALLAH*
- ❖ *Mr. BELABED .AMINE*

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

# 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 Mes frères et sœurs.*
- *A Mes responsables au vice rectorat de la pédagogie Mr Lahfa Farid et Mr Sabagh Djamel*
- *A tous les membres de ma promotion de MID.*
- *A mes amis sans exception surtout ceux du vice rectorat de la pédagogie.*
- *A tous mes professeurs.*

*Boufatah Abdesselam*

# 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 et à mon père, que dieu les récompense et les garde, et surtout ma mère qui m'a éclairé mon chemin et qui m'a encouragé et soutenu toute au long de mes études.*
- *A Mes frères et sœurs.*
- *A Mes responsables au Faculté des sciences économiques et sciences des commerces et de la gestion Mr Bendi Abdessalam et Mr Moutas, et les responsables de la Faculté des sciences.*
- *A tous les membres de ma promotion de MID.*
- *A mes amis sans exception surtout ceux du Faculté des sciences économiques et sciences des commerces et de la gestion.*
- *A tous mes professeurs.*

*Belaidi Soufiane...*

## Résumé

Les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications sur l'Internet. Ils constituent la technologie de base pour le développement d'architectures orientées services. Ces architectures sont de plus en plus répandues sur le Web. Le principe essentiel du modèle de service Web est de transformer le Web en un dispositif distribué d'échange et de calcul, où les services Web peuvent interagir d'une manière intelligente. Actuellement, de nombreux services Web, avec des fonctionnalités similaires sont fournis par des fournisseurs concurrents, un problème qui apparaît à la mise en œuvre de cette technologie est que le processus de sélection devient assez complexe.

Notre travail consiste à sélectionner, une composition de services à base de recherche Taboue .cette approche prend en compte les critères de QOS, pour instancier une composition abstraite.

**Mots Clés :** Services Web, Architectures Orientées Services, Sélection, recherche Taboue.

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 work an approach based on tabu search. 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, Quality of services, Optimization, Tabu Search.

ملخص

---

## TABLE DE MATIERE

<b>INTRODUCTION GENERALE.....</b>	<b>5</b>
<b>CHAPITRE I LES SERVICES WEB.....</b>	<b>9</b>
<b>1. INTRODUCTION.....</b>	<b>10</b>
<b>2. L'ARCHITECTURE ORIENTEE SERVICE SOA.....</b>	<b>11</b>
2.1. DEFINITION.....	11
2.2. LES AVANTAGES DE SOA.....	12
<b>3. LES SERVICES WEB.....</b>	<b>12</b>
3.1. DEFINITION.....	12
3.2. L'INTERET DES SERVICES WEB.....	13
3.3. LES CARACTERISTIQUES DES SERVICES WEB.....	14
3.4. L'ARCHITECTURE DES SERVICES WEB.....	14
3.5. LES PRINCIPALES TECHNOLOGIES DE DEVELOPPEMENT DE SERVICE WEB.....	16
3.5.1. <i>Communication</i> .....	16
3.5.2. <i>Description</i> .....	20
3.5.3. <i>Publication</i> .....	22
3.6. LES AVANTAGES ET INCONVENIENT DES SERVICES WEB.....	23
3.6.1. <i>Avantages</i> .....	23
3.6.2. <i>Inconvénients</i> .....	23
<b>4. SELECTION DE SERVICE WEB.....</b>	<b>24</b>
<b>5. CONCLUSION.....</b>	<b>26</b>
<b>CHAPITRE II CONCEPTION ET IMPLEMENTATION.....</b>	<b>27</b>
<b>1. INTRODUCTION.....</b>	<b>28</b>
<b>2. L'ALGORITHME TABOU.....</b>	<b>28</b>
2.1. HISTORIQUE ET DEFINITION.....	28
2.2. PRINCIPE DE BASE.....	29
2.2.1. <i>L'algorithme de la RT en générale</i> .....	30
2.2.2. <i>Critère d'arrêt dans la RT</i> .....	31
2.2.3. <i>Comportement de l'algorithme tabou</i> .....	32
2.2.4. <i>Diverses améliorations de la RT</i> .....	33
2.2.5. <i>Domaines concernés par la RT</i> .....	33
2.3. L'ALGORITHME DE LA RT AVEC REDEMARRAGE ALEATOIRE.....	34

---

<b>2.4.</b>	<b>DESCRIPTION DE LA BASE</b> .....	<b>34</b>
<b>2.5.</b>	<b>DESCRIPTION DE LA BASE [QU ET BOUGUETAYA, 2010]</b> .....	<b>35</b>
<b>2.6.</b>	<b>DESCRIPTION DE LA REQUETE</b> .....	<b>35</b>
<b>2.7.</b>	<b>LA FONCTION OBJECTIF « SCORE OU FITNESS »</b> .....	<b>35</b>
<b>3.</b>	<b>CONCEPTION</b> .....	<b>36</b>
3.1.	DIAGRAMME DE CAS D'UTILISATION .....	36
3.2.	DIAGRAMME DE SEQUENCE .....	37
3.3.	DIAGRAMME DE CLASSES .....	38
<b>4.</b>	<b>INTERFACE HUMAIN/MACHINE (IHM)</b> .....	<b>39</b>
<b>5.</b>	<b>EXPERIMENTATION</b> .....	<b>41</b>
<b>6.</b>	<b>CONCLUSION</b> .....	<b>42</b>
	<b>CONCLUSION GENERALE</b> .....	<b>44</b>
	<b>ANNEXE</b> .....	<b>46</b>
	<b>REFERENCES</b> .....	<b>49</b>

---

## LISTE DE FIGURES

FIGURE I. 1: ARCHITECTURE ORIENTEE SERVICES (SOA).....	122
FIGURE I. 2: ARCHITECTURE DU SERVICE WEB .....	166
FIGURE I. 3 : EXEMPLE FICHIER XML DE BIBLIOTHEQUE.....	177
FIGURE I. 4 : STRUCTURE D’UN MESSAGE SOAP .....	ERREUR ! SIGNET NON DEFINI.9
FIGURE I. 5 : STRUCTURE D’UN DOCUMENT WSDL.....	221
FIGURE I. 6 : LES TROIS TYPES DE L’ANNUAIRE UDDI.....	22
FIGURE I. 7 : EXEMPLE DE SELECTION (AGENCE DE RESERVATION) .....	244
FIGURE I. 8 : EXEMPLE DE MOTIVATION .....	25
FIGURE I. 9 : LES APPROCHES DE SELECTION.....	255
FIGURE II. 1 : L’ORGANIGRAMME DE PRINCIPE DE TABOU .....	Erreur ! Signet non défini.1
FIGURE II. 2 : DIAGRAMME DE CAS D’UTILISATION.....	37
FIGURE II. 3 : DIAGRAMME DE SEQUENCE.....	38
FIGURE II. 4 : DIAGRAMME DE CLASSE.....	ERREUR ! SIGNET NON DEFINI.9
FIGURE II. 5 : INTERFACE PRINCIPALE .....	ERREUR ! SIGNET NON DEFINI.
FIGURE II. 6 : VALIDATION DES CONTRAINTES.....	37
FIGURE II. 7 : LANCEMENT DE SELECTION.....	438
FIGURE II. 9 : HISTOGRAMME D’OPTIMALITE POUR LA RECHERCHE TABOUE STANDAR .....	41
FIGURE II. 9 :HISTOGRAMME D’OPTIMALITE POUR LA RECHERCHE TABOUE AMELIOREE .....	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.

RT : Recherche Tabou.

# Introduction générale

### **Contexte**

La majorité des organisations actuellement se tournent vers des architectures à base de services Web pour le développement et l'intégration d'applications ou de systèmes d'informations. L'importance des standards dans ce contexte a sans doute accentué le phénomène. Mais il arrive très fréquemment que plusieurs services répondent à un même ensemble de besoins fonctionnels, d'où la nécessité de la naissance du domaine de la sélection de services Web à base de QoS (qualité de service).

Les services Web (en anglais Web services) représentent un mécanisme de communication entre applications distantes à travers le réseau internet indépendant. Telle que ces applications peuvent être vues comme un ensemble de services métiers, structurés et correctement décrits, dialoguant selon un standard international plutôt qu'un ensemble d'objets et de méthodes entremêlés.

Les services Web considérés comme une fonction accessible à distance par une requête qui exprime les besoins d'un utilisateur a pour entrer un ensemble de concepts indiqués dans une procédure spécifique et donne comme sortie un autre ensemble de concepts.

Le problème de la recherche dans les services Web a attiré l'attention des chercheurs au cours de la dernière décennie. La raison de cela est que la technologie évolue et que beaucoup de services sont disponibles, il devient important d'être en mesure de localiser les services qui répondent à nos besoins dans un grand nuage dense d'offres.

Plusieurs propositions ont été avancées pour résoudre ce problème et plusieurs normes ont été définies, mais aucun d'entre eux a été efficace ou est maintenant acceptée comme le moyen d'effectuer une recherche de service.

Ce mémoire se penche sur une perspective plus large, une solution concrète et applicable.

### **Problématique**

Aujourd'hui le besoin d'un client est de nature complexe (minimiser les coûts, minimiser les délais, augmenter le taux de service...). De plus, comme les services Web

avec des fonctionnalités similaires sont censés être fournis par des fournisseurs concurrents, Il sera donc nécessaire, de fournir un outil d'aide à la décision.

La qualité de service dans le cas des services web se mesure à l'aide de plusieurs métriques dont les métriques de performance, de disponibilité et de fiabilité. Une recherche sur internet nous permet certainement de trouver plusieurs services web qui remplissent ces critères. Mais lequel sera le plus adéquat, le moins cher, le plus luxueux, le plus rapide, etc. Lequel de ces services sera le plus disponible, le plus fiable? Lequel aura un temps de réponse *acceptable*? Il devient ainsi nécessaire de choisir les services web pertinents parmi ceux trouvés et de fixer des critères pour choisir les meilleurs.

L'objectif de ce travail est de sélectionner (de façon proche de l'optimale) des services web à base de leurs qualités de services.

### **Contribution**

Pour répondre à cet objectif nous préconisons l'utilisation de la recherche taboue et de bénéficier de ses avantages pour permettre de sélectionner des services web adaptés aux préférences des clients.

Les algorithmes tabous sont répandus dans divers domaines pour résoudre des problèmes d'optimisation et de recherche.

### **Plan du mémoire**

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

#### **Chapitre I**

Ce chapitre est consacré à l'étude des services Web et leur composition pour mieux comprendre les concepts de base de cette technologie. Ce chapitre se divise en deux parties : nous présentons dans la première partie les services Web, leur architecture, leur modèle d'interaction et leurs standards de base. Dans la deuxième partie de ce dernier, nous aborderons la composition des services Web, en suite quelques avantages et inconvénients.

#### **Chapitre II**

Ce chapitre est consacré à la conception, l'implémentation et l'expérimentation du prototype. Au début nous présentons notre algorithme Tabou, et une terminologie pour définir le vocabulaire employé par notre travail. Enfin nous avons présenté la conception de notre système et l'implémentation pour évaluer la performance de la méthode proposée.

#### **Conclusion générale**

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

# Chapitre I

## Les services Web

## 1. Introduction

De nos jours, il devient important de faire communiquer entre les différents systèmes d'informations des entreprises pour échanger des informations et des services. Ce type de communication est appelé intégration B2B ; qui se réfère à « la technologie de logiciel utilisée pour connecter n'importe quel système d'information d'une organisation à tous ses associés de commerces ». L'augmentation De nombreuses organisations se tournent vers les services Web à base d'architecture pour le développement et l'intégration d'applications et de systèmes d'information. Les dernières décennies ont vu le développement rapide des systèmes d'information distribués, notamment 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 architecture logicielle s'appuyant sur un ensemble de services simples. L'objectif d'une architecture orientée services est donc de décomposer une fonctionnalité en un ensemble de fonctions basiques, appelées **services**, fournies par des composants et de décrire finement le schéma d'interaction entre ces services. Donc SOA est une approche de conception basée sur des standards (SOAP, WSDL, ...), afin de créer une infrastructure intégrée, capable de répondre rapidement à l'évolution des besoins d'un utilisateur. Elle fournit des principes et directives pour la transformation d'un réseau intégrée des ressources complexe et rigide existant de ressources informatiques hétérogènes, distribués, complexes notamment simplifié et flexible qui peut être modifié et combiné à mieux répondre aux objectifs de l'utilisateur. [1]

Dans ce chapitre, nous allons présenter l'architecture orientée services et sa principale réalisation: les services Web. Nous donnons les définitions de services Web et décrire brièvement leurs normes et principes des technologies. Enfin, nous discutons de la notion de composition de services web en expliquant les méthodes de sélection des services Web.

## 2.L'architecture orientée service SOA

### 2.1. Définition

Plusieurs définitions sont utilisées pour définir et expliquer la SOA. Les définitions qui suivent illustrent différentes vues de la SOA. Cependant, elles convergent toutes à une seule direction :

- **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].*

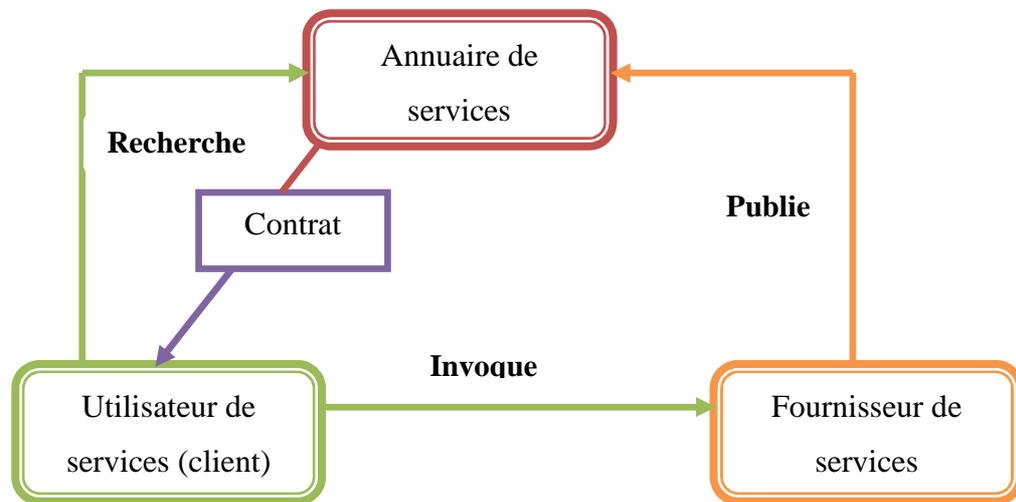
- **Technique :**

*« 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, une SOA (Architecture Orientée Services, Service Oriented Architecture) basée sur la réorganisation des applications à partir d'un ensemble de services de base. Ces applications (visible) représentent les services en question en fonction parlant fonctionnellement comme des interfaces standards (*langages de Representational State Transfer* REST ou SOAP / WSDL), connue sous le nom de services Web couches potentiellement compréhensibles par tous les systèmes concernés, pour autant qu'ils comprennent le module d'interprétation nécessaire. Dans un tel environnement, les services (appelés «producteurs») sont ainsi exposés à d'autres services (appelés «consommateurs»). [4]

Comme le montre la figure ci-dessous, les fournisseurs de services sont publier leurs services dans un répertoire de services ou annuaire de service pour répondre à la SOA, les clients ou les usagers afficher ce répertoire pour trouver les services qui satisfont les critères qui correspondent certaines descriptions, si elle est cochée le répertoire répond aux clients en fournissant une description de l'application d'un

accord de l'utilisateur. Ensuite, le client fait son choix en communiquant avec le fournisseur pour invoquer le service Web. [5]



**Figure I. 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

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 complexes. Une fois qu'un service Web est déployé, d'autres applications peuvent le découvrir et l'invoquer.

« Un service web est une agrégation de fonctionnalités publiées pour être utilisées. Il utilise internet comme conduit pour réaliser une tâche. Il est semblable à un processus métier virtuel qui définit des interactions au niveau application ».

« 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 complexes. Une fois qu'un service Web est déployé, d'autres applications peuvent le découvrir et l'invoquer ».[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 les applications. Ainsi, les applications utilisant des technologies différentes peuvent envoyer et recevoir des données à travers compréhensible par tous les protocoles. Les services Web sont normalisés, car ils cherchent généralement des standards XML et HTTP pour le transfert de

données et ils ont une affinité avec de nombreux autres environnements de développement. Ils sont exploitables à distance via n'importe quel type de plateforme. C'est dans ce contexte qu'un intérêt tout particulier a été apporté à la conception de services Web, car elles permettent aux entreprises d'offrir des applications rentables à distance par d'autres sociétés. [10]

### 3.3. Les caractéristiques des services Web

Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre de nombreuses d'opportunités pour les développeurs web qui ont les caractéristiques suivantes : [11]

- **Modularité** : le modèle Service Web est modulaire, on peut facilement réutiliser et combiner des Services Web déjà existant pour former un nouveau Service Web.
- **Interopérabilité** : Un Service Web est indépendant de la plateforme et du langage de programmation avec lesquels il est développé. , 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.
- Il est accessible via le réseau.
- Il dispose d'une interface publique (ensemble d'opérations) décrite en XML ;
- Ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire.

En plus de ces avantages, la technologie des services web est soutenu par des grands acteurs de l'industrie informatique tels que : IBM, HP, Microsoft et le consortium W3C.

### 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 :

**Architecture de base :** est une architecture de référence, elle contient trois couches principales, qui sont :

- Le fournisseur de service : c'est le propriétaire du service.
- Le client (ou le consommateur de service) : c'est un demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service.
- L'annuaire des services : c'est un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.

**Architecture en couche:** est une architecture est plus complète, elle utilise les couches standards de la première architecture en ajoutant au-dessus d'autres couches plus spécifiques. Elle est appelé architecture étendue ou encore en Pile. [12]

Cette architecture est composée de quatre grandes couches mettant en œuvre plusieurs technologies, chaque couche s'appuyant sur un standard particulier : [13]

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

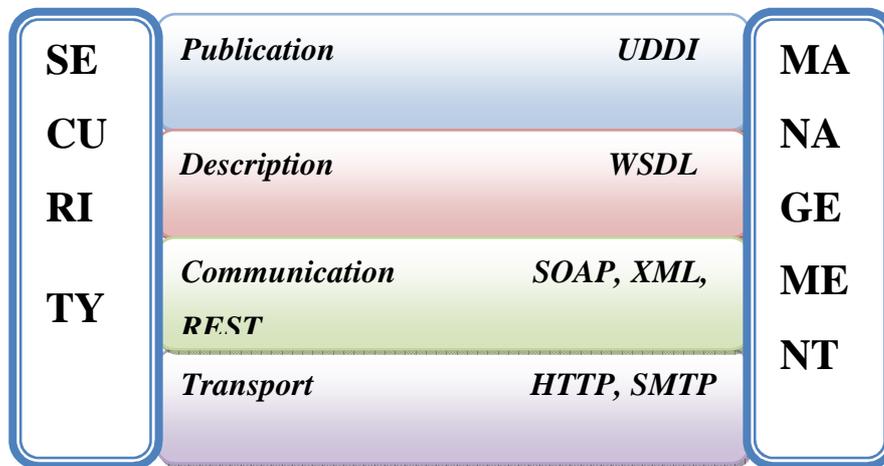


Figure I. 2: Architecture du service Web

### 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 le langage destiné à succéder à HTML sur le « World Wide Web ». Comme le « HTML », c'est un langage de balisage (Markup), c'est-à-dire un langage qui présente de l'information encadrée par des balises. Contrairement à HTML, qui présente un jeu limité de balises orientées présentation (titre, paragraphe, image, lien hypertexte), XML est un métalangage qui va permettre d'inventer à volonté de nouvelles balises pour isoler toutes les informations élémentaires (titre d'ouvrage, prix d'article, numéro de sécurité sociale, référence de pièce, ...) ou agrégats d'informations 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 fermante (<, >), 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]

```
<MI>
  <Math>..... </Math>
  <Informatique year="2013">
    <Licence>
      <L2>Stevens</L2>
      <Spécialite>Informatique</ Spécialite >
      <L3>Stevens</L3>
      <Spécialite>SI</ Spécialite >
      <Spécialite>RSD</ Spécialite >
    </ Licence >
    <Master>
      <M1>
        <Spécialite>MID </ Spécialite >
        <Spécialite>SIC </ Spécialite >
        <Spécialite>RSD </Spécialite >
      </M1>
      <M2>
        <Spécialite>MID </ Spécialite >
        <Spécialite>SIC </ Spécialite >
        <Spécialite>RSD </Spécialite >
      </M2>
    </ Master >
  </ Informatique >
</MI>
```

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

## ▪ REST

REST est une architecture de services Web, à la manière de SOAP et de XML-RPC. C'est l'acronyme de Representational State Transfer. Elaboré en l'an 2000 par Roy Fielding, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd et d'autres travaux fondamentaux, REST est à l'origine une tentative de décrire les principes de l'architecture du Web. [17]

## ▪ Le Protocol SOAP

SOAP (*simple Object Access Protocol*) est un protocole de communication définit 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 repartie, 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 est indépendant de tout système d'exploitation et n'importe quel langage de programmation, car il utilise le langage XML. Cela permet à l'exposition et la consommation de services Web avec différents outils et systèmes d'exploitation.

SOAP définit un format pour envoyer des messages. Des messages SOAP sont structurés dans un document XML et deux parties obligatoires: l'enveloppe SOAP et le corps du message SOAP, et une partie optionnelle: l'en-tête SOAP.

### ✓ Structure d'un message SOAP

Un message SOAP est contenu dans une enveloppe, ainsi le tag racine d'un document SOAP est le tag <Envelope>. Ce message est basé sur le langage XML (Figure 1.4).

La structure d'une enveloppe SOAP se compose de plusieurs parties :

- Un entête optionnel composé d'un ou plusieurs headers : elle contient des informations sur le traitement du message
- Un corps (Body) : il contient les informations de la requête ou de la réponse
- Une gestion d'erreurs optionnelle (Fault) contenue dans le corps.
- Des pièces jointes optionnelles (attachment) contenues dans le corps.



**Figure I.4 : Structure de message SOAP.**

**L'en-tête du protocole de transport:** qui dépend de protocole de transport utilisé, par exemple si le protocole HTTP est utilisé, l'en-tête contient :

- La version de protocole HTTP utilisée.
- La date de génération de message SOAP.
- Le type d'encodage du contenu (généralement de type XML).

**L'enveloppe SOAP :** la partie principale d'un message SOAP est l'enveloppe (symbolisée par la balise enveloppe), cette dernière est subdivisée en deux sous-parties: la partie en-tête (Header) et la partie corps du message (Body).

**L'en-tête SOAP:** est optionnelle et extensible. Les balises XML qui permettent de symboliser cette partie sont: `<env:Header>` et `</env:Header>`. Ces balises peuvent être complétées par des attributs permettant de définir le domaine de noms du service Web. En fait, l'en-tête permet principalement d'ajouter des informations sur le comportement des différents noeuds intermédiaires, lors de traitement du message.

Un noeud étant un intermédiaire SOAP, incluant le récepteur et l'émetteur SOAP, désignable depuis un message SOAP. Son rôle est de traiter l'en-tête, ensuite de transférer le résultat (le message SOAP modifié) à un autre intermédiaire (qui peut être le récepteur final).

**Le corps SOAP:** l'élément corps de message SOAP contient des données spécifiques à l'application distante. Les balises symbolisant cette partie sont: **<env:Body>**et **</env:Body>**. Les données doivent donc être sérialisées selon l'encodage XML. En plus, des données de cette partie peuvent transporter un type spécial comme: les messages d'erreurs (SOAP Fault). Le contenu du Corps SOAP est normalisé dans SOAP RPC, pour modéliser une requête et sa réponse. Le body de la requête contient l'identifiant de l'objet distant, le nom de la méthode à exécuter et les éventuels paramètres. Le body de la réponse contient le résultat de l'exécution de la requête.

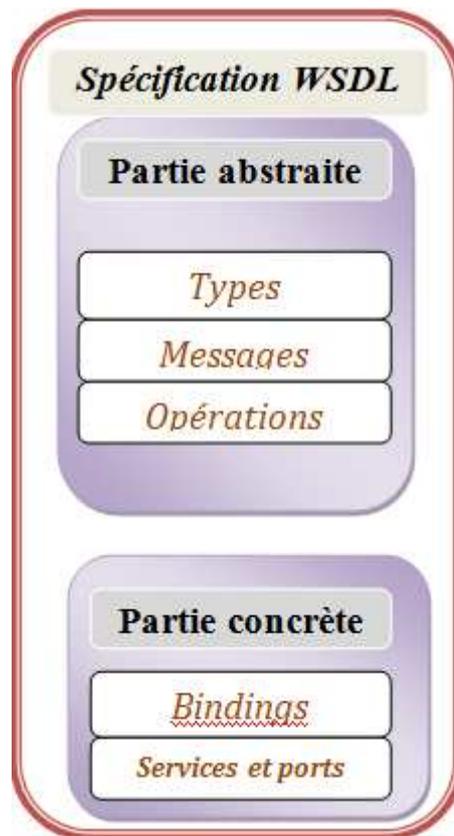
### **3.5.2. Description**

- **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 I. 4 : Structure d'un document WSDL**

En résumé un fichier WSDL contient donc sept éléments.

- **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.
- **PortTypes** : 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.
- **Opération** : c'est la description d'une action exposée dans le port.

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 (acronyme d'Universel *Description, Discovery and Integration*) est un annuaire de services fondé sur XML et plus particulièrement destiné aux services Web. Il est utilisé pour fournir une description d'un service web afin de permettre son utilisation. C'est une recommandation du W3C. permet de localiser sur le réseau le service Web recherché ; car il permet l'accès aux répertoires des utilisateurs potentiels de services web.

UDDI a été conçu en 2000 à l'initiative lancée par *ARIBA, Microsoft* et *IBM*, pour devenir le registre standard de la technologie des services Web. Pour répondre aux services Web de technologie, référencés dans UDDI sont accessibles via le protocole de communication SOAP, et la publication d'informations concernant les fournisseurs et les services doit être spécifiée en XML afin que la recherche et l'utilisation soient dynamiquement et automatiquement. UDDI est une spécification mise au point par l'OASIS définissant la manière de publier et découvrir des services Web sur un réseau. [22]

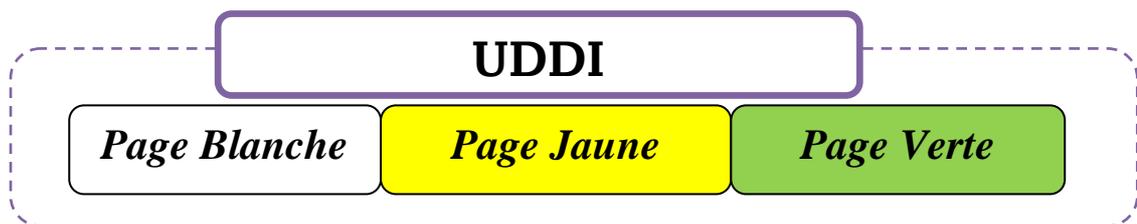


Figure I. 5 : les trois types de l'annuaire UDDI

Les informations qu'il contient peuvent être organisées selon trois méthodes :

- **Les pages blanches** : annuaire classique par nom, incluant l'adresse, le contact et les identifiants relatifs aux services Web (sur les entreprises)
- **Les pages jaunes** : annuaire thématique, identifiant les secteurs d'affaires relatifs aux services Web. (organisé selon une taxonomie).

- **Les pages verts:** annuaire technique; une fois ceci est fait, le service Web peut alors être connu de tous ceux qui le recherchent. Le modèle Un registre UDDI se compose de quatre types de structures de données, le businessEntity, le businessService, le bindingTemplate et la tModel. Cette répartition par type fournit des partitions simples pour faciliter la localisation rapide et la compréhension des différentes informations qui constituent un enregistrement. [23]

### 3.6. Les avantages et inconvénient 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

- Les normes de services Web dans certains domaines sont actuellement récentes.

- 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 informatique a considérablement évolué depuis ces années. L'émergence de l'approche des services a été accompagnée par une augmentation exponentielle du nombre de services et les fournisseurs. Avec la sélection de services web, nous essayons de choisir le meilleur fournisseur d'un service Web, étant donné un ensemble de fournisseurs de ce service.

Par exemple, une agence de voyage nécessite la consommation de plusieurs types de services pour satisfaire le besoin de ses clients: une réservation pour les hôtels, une location de voiture.

Le client veut que l'hôtel soit plus proche de la ville de Tlemcen à un prix plus avantageux.....

En plus le cout total de la composition doit être inférieur à une limite.

Ce type de problème est appelé sélection orientée QOS.

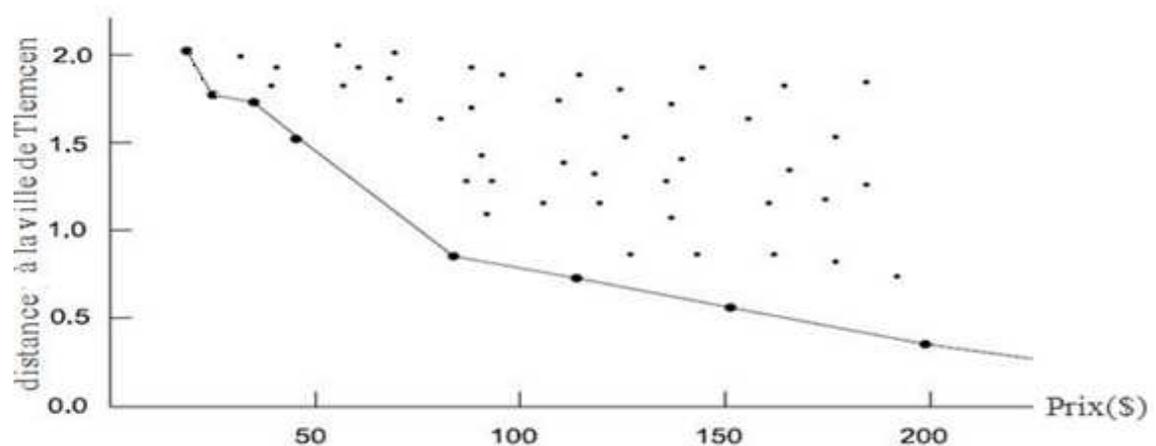


Figure I. 6 : Exemple de sélection (agence de réservation)

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

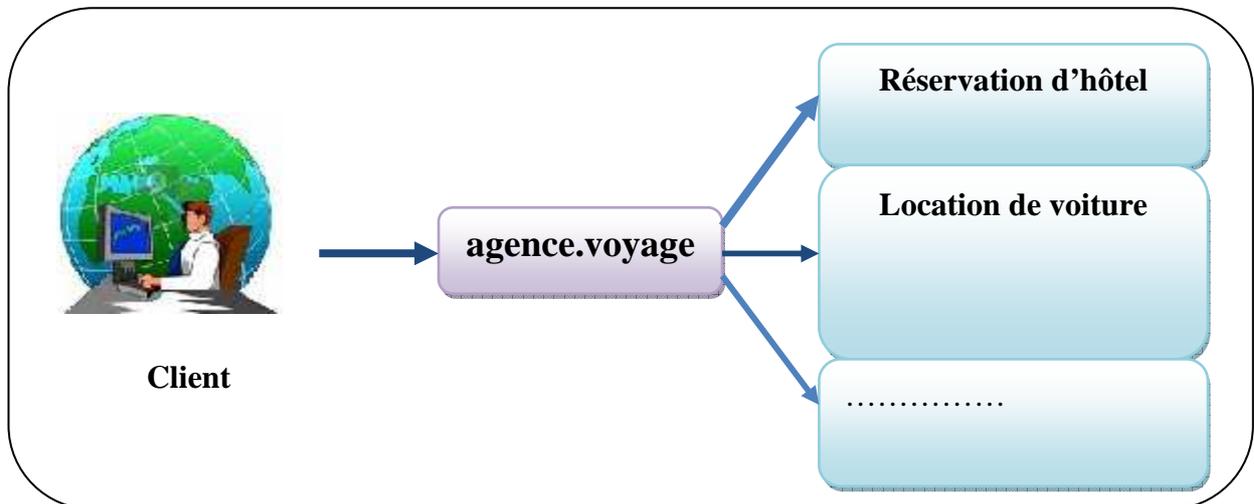


Figure I. 7 : Exemple de motivation.

### ✓ 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.

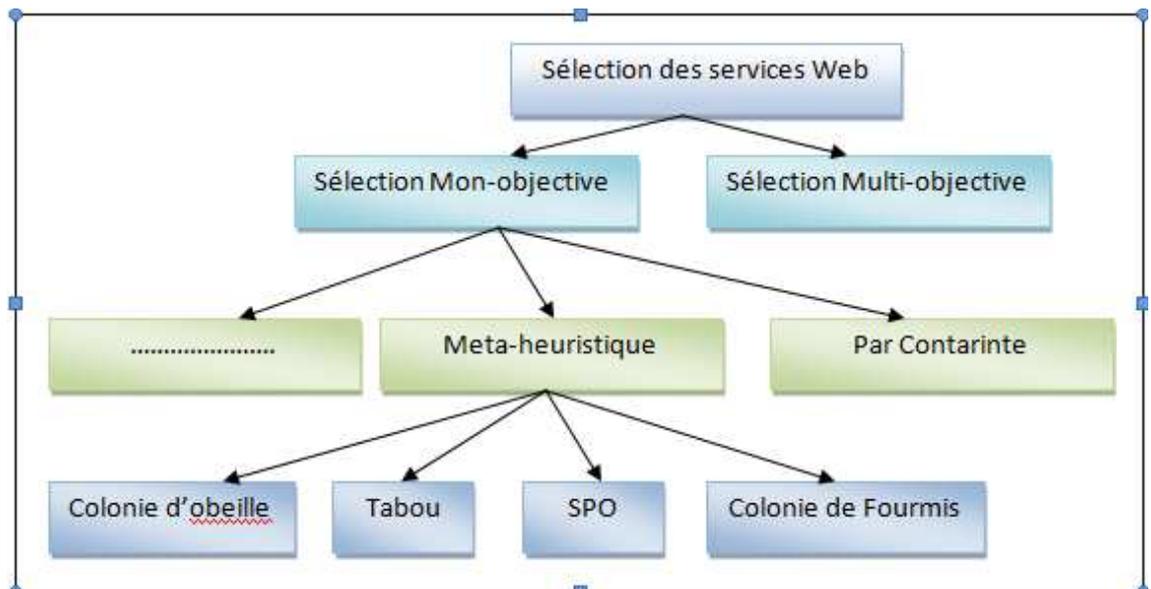


Figure I.8 : Les approches de sélection.

## **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 II

## Conception et Implémentation

### **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 la recherche Tabou, en anglais : Search Tabou.

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 Tabou (origine, principe...), puis nous présenterons une description de la base, après nous allons parler du processus de développement logiciel utilisé, en suite, nous finissons par présenter les différents diagrammes de conception ainsi que l'application et les résultats obtenus.

### **2. L'Algorithme Tabou.**

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

La recherche avec tabou a été proposée par Fred Glover en 1986. Depuis cette date, la méthode est devenue très populaire, grâce aux succès qu'elle a remportés pour résoudre de nombreux problèmes.

La Méthode Tabou est une classe des métaheuristiques de recherche locale utilisée pour résoudre des problèmes complexes et/ou de très grande taille (souvent NP-durs). La RT a plusieurs applications en programmation non linéaire (PNL). La révolution de cette méthode par rapport aux autres: permet de surmonter le problème des optima locaux par l'utilisation de listes taboues (principe de mémoire).

C'est une méthode à mémoire adaptative :

- ✓ Mémoire à court terme : diversification.
- ✓ Mémoire à long terme : intensification.

Cette méthode est généralement meilleure en temps d'exécution.

### **2.2. Principe de base**

La recherche taboue se base sur l'utilisation de structures de mémoire, qui permette de mener une recherche plus "intelligente" dans l'espace des solutions. La méthode de recherche Tabou fonctionne avec une seule configuration courante, qui est actualisée au cours des itérations successives. [27]

Le principe de base consiste à choisir la meilleure solution du voisinage de la solution courante, parfois moins bonne que la solution courante. Par conséquent, la recherche ne s'arrête pas au premier optimum local trouvé. Le danger est alors de revenir à des solutions déjà explorées. Pour éviter des cycles, on mémorise les dernières solutions visitées dans une liste taboue et on interdit tout mouvement qui conduit à une solution de cette liste. La liste taboue est donc une sorte de mémoire à court terme. Tout mouvement qui nous mène de la solution courante à une solution de la liste tabou est appelé mouvement tabou. Lorsque la liste tabou est trop complexe ou occupe une grande place de mémoire, on mémorise plutôt des attributs des solutions ou des modifications des solutions que des solutions entières. La mémorisation de ces attributs ou modifications permet certes un gain en place mémoire, mais elle n'a pas que des avantages. [28]

Poursuivre la recherche de solutions même lorsqu'un optimum local est rencontré,

- En permettant des déplacements qui n'améliorent pas la solution.
- En utilisant le principe de mémoire pour éviter les retours en arrière (mouvements cycliques).

#### ▪ **Mémoire**

- Elle est représentée par une liste taboue qui contient les mouvements qui sont temporairement interdits.
- Mouvements interdits ou solutions interdites.
- Son rôle évolue au cours de la résolution: diversification (exploration de l'espace des solutions) vers intensification.

▪ **Exception aux interdictions :**

Il est possible de violer une interdiction lorsqu'un mouvement interdit permet d'obtenir la meilleure solution enregistrée jusqu'à maintenant.

**2.2.1. L'algorithme de la RT en générale**

Engendrer une configuration initiale  $S_0, S := S_0$

$S^* := S ; f^* := f(S)$  ( $S$  est choisi de façon aléatoire)

$T := \{ \}$  // liste taboue

Répéter

$m :=$  le meilleur mouvement parmi les mouvements non tabous et les mouvements tabous exceptionnels (critère d'aspiration)

$S := S (+) m$

Si  $f(S) < f(S^*)$  faire  $S^* := S ; f^* := f(S)$

Mettre  $T$  à jour ;

Jusqu'à <condition fin **ou** critère d'arrêt >

Retourner  $S^*$ . [29]

- **Caractéristiques de l'algorithme:** Voici l'explication des différentes variables du schéma :

**$S_0$**  : solution initiale.

**$S^*$**  : meilleure solution jusqu'à présent.

**$S'$  et  $S$**  : nouvelles solutions du voisinage de  $S^*$ .

**$f(x)$**  : fonction objectif à minimiser.

**$f^* = f(s^*)$**  : valeur de la meilleure solution. [29]

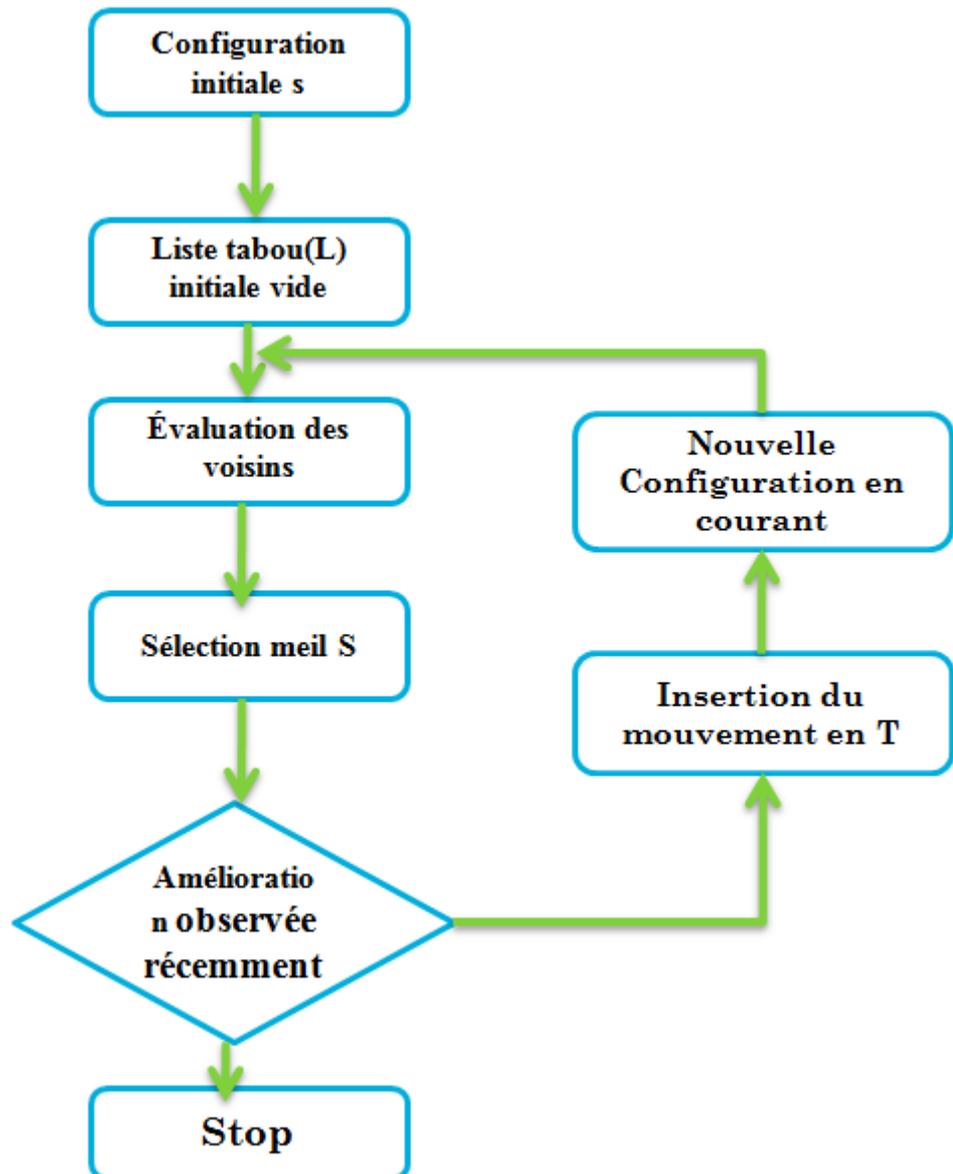


Figure II. 1 : L'organigramme de principe de Tabou.

### 2.2.2. Critère d'arrêt dans la RT

On peut arrêter la recherche à tout moment contrairement au recuit simulé, par des critères d'arrêt possibles comme :  Si une solution prouvée optimale a été trouvée.

- Si une limite a été atteinte en ce qui concerne :
  - ✓ Le nombre d'itérations.
  - ✓ Le temps de calcul.
- Si la recherche semble stagner : nombre d'itérations sans amélioration de la meilleure configuration trouvée.

Dans la recherche avec tabou, on cherche «normalement» le meilleur mouvement non tabou. Cependant, cette manière de procéder peut se révéler trop coûteuse.

Pour y remédier, on peut se limiter à engendrer seulement un sous-ensemble des voisins (Ou mouvements). On parle de liste de candidats :

- **Construction de la liste de candidats :**
  - ✓ Les candidats de la liste peuvent être engendrés de manière aléatoire.
  - ✓ Une autre approche consiste à sélectionner les voisins qui semblent les plus prometteurs – selon un critère quelconque.

### 2.2.3. Comportement de l’algorithme tabou

- **Si la liste taboue est courte : il y a moins d’interdictions (mouvements tabous).**
  - ✓ La recherche épouse mieux les optima locaux rencontrés.
  - ✓ L’algorithme tend à parcourir de moins grandes distances dans l’espace de recherche. Il explore moins l’espace de recherche.
  - ✓ Le risque de cycles est plus grand.
- **Si la liste taboue est longue :**
  - ✓ Il y a avantage d’interdictions (mouvements tabous).
  - ✓ La recherche risque de manquer de nombreux optima locaux sur son chemin.
  - ✓ L’algorithme tend à parcourir de plus grandes distances dans l’espace de recherche.
  - ✓ Il explore davantage l’espace de recherche.
  - ✓ Le risque de cycles est réduit.

Le comportement de l’algorithme dépend :

- ✓ De la longueur de la liste taboue.
  - ✓ Mais aussi de la taille de la liste de candidats (si on en utilise une).
- **Dans le cas où une liste de candidats est utilisée :**
    - ✓ Les meilleurs mouvements risquent de ne pas appartenir à la liste. Il y a donc deux «forces» qui conjuguent leurs effets pour ressortir des optima locaux : la liste taboue et la liste de candidats.

- ✓ Plus la liste de candidats est petite, moins la liste taboue a besoin d'être grande.

### 2.2.4. Diverses améliorations de la RT

La liste taboue peut s'avérer trop contraignante lors de la recherche d'une solution. Le mécanisme d'aspiration permet de lever ponctuellement le statut "tabou" afin d'atteindre des solutions inédites.

- **L'intensification** : est l'une des stratégies qui permet de mémoriser les meilleures solutions rencontrées (ou leur configuration) et les utilise afin d'améliorer la recherche.
- **La diversification** : cherche à utiliser des mouvements encore jamais réalisés afin d'explorer des régions nouvelles de l'espace de recherche en mémorisant bien sur les solutions les plus visitées. bien sûr, les deux stratégies sont complémentaires.
- Sélection du meilleur voisin :
  - ✓ **Best Fit** : le voisinage est exploré en entier.
  - ✓ **First Fit** : une partie du voisinage est explorée.
- Utilisation d'une table de calculs : pour éviter de calculer entièrement le coût de chaque voisin, à chaque itération on mémorise dans une table les modifications au coût de la solution courante associées à chacun des mouvements possibles

### 2.2.5. Domaines concernés par la RT

- Problèmes de transport.
- Planification et ordonnancement.
- Optimisation de graphes.
- Télécommunications.
- Logique et intelligence artificielle.
- Création d'horaires.
- Optimisation de structures.
- Techniques spécialisées.

### 2.3. L'algorithme de la RT avec redémarrage aléatoire

Engendrer une configuration initiale  $S_0$  ;  $S := S_0$  (un vecteur ayant 10 entiers)

1-  $f^* := f(S)$

2-  $S^* := S$  ; ( $S$  est choisi de façon aléatoire)

3-  $T := \{ \}$  // liste taboue

4-  $t = 0$ ;

5- Répéter

$m :=$  le meilleur mouvement parmi les mouvements non tabous et les mouvements tabous exceptionnels (critère d'aspiration) (une permutation de deux classes)

$S := S (+) m$

Si  $f(S) < f(S^*)$  faire  $S^* := S$  ;  $f^* := f(S)$

Mettre  $T$  à jour ;

Si  $f(s^*)/f(\text{sol-optimal}) < 50\%$  et  $(t > 40\% * T_{\max})$  Alors aller à (2)

Jusqu'à  $t = T_{\max} + 1$  ou critère d'arrêt >

6- Retourner  $S^*$ .

#### ▪ Discussion

Cette amélioration permet d'éviter les régions non prometteuse, et surtout lorsqu'on dépasse un certain nombre d'itération.

La section d'expérimentation offre une comparaison entre la version standard, et la version randomisée.

### 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 cinq (5) paramètres pour évaluer les opérations des services : délai, fiabilité, disponibilité, coût 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 générées

en se basant sur une distribution uniforme. Un échantillon de notre base est illustré dans la table suivant 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 base [Qu et bouguetaya, 2010]

## 2.6. 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

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

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

Qualité(Si) = (latence(Si), fiabilité(Si), disponibilité(Si), coût(Si), réputation(Si)).

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

Le but de fonction de score c'est pour calculer une valeur hors du vecteur de QoWS des services. Cela peut faciliter la comparaison de la qualité des services. Comme les utilisateurs peuvent avoir des préférences sur la façon dont leurs requêtes sont traitées.

Fctobj(sol)=(Cout(sol), Rep(sol), Disp(sol), Fiab(Sol), Temp(sol))

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

(sol)

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

Avec :

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

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

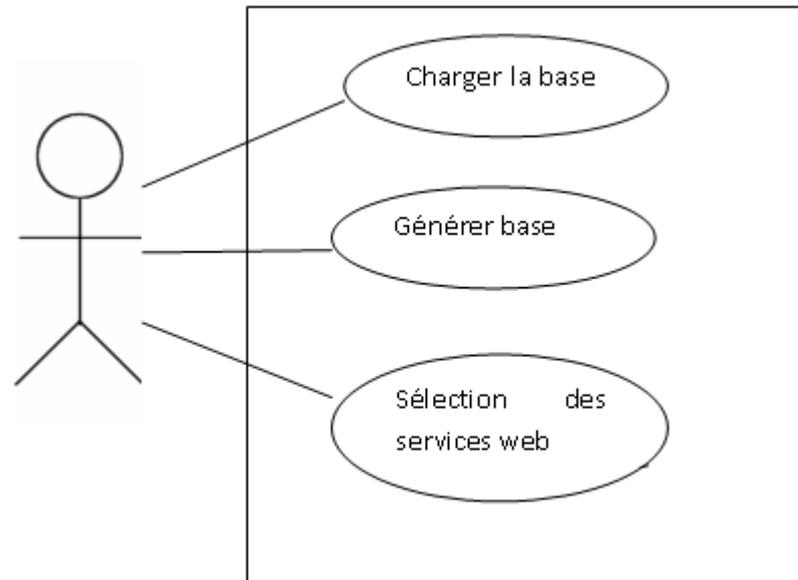
Sol : est un vecteur contenant 10 entiers, chacun d'eux représente l'instance de service choisie pour une classe abstraite donnée.

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.  $Q_i^{\max}$  est la valeur maximale pour l'ième paramètre QoWS pour tous les plans possibles d'exécution du service et  $Q_i^{\min}$  est le minimum.

### 3. Conception

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

Un diagramme de cas d'utilisation (en anglais use cases) est constitué une technique qui détermine les besoins des utilisateurs et capter les exigences fonctionnelles d'un système. En d'autre terme, il décrit le comportement d'un système du point de vue des ses utilisateurs. Il décrit aussi les interactions entre les utilisateurs d'un système et le système lui-même. La **Figure II.2** représente le diagramme de cas d'utilisation de notre application.



**Figure II. 2: Diagramme de cas d'utilisation.**

### **3.2. Diagramme de séquence**

Un diagramme de séquence permet de des représenter les interactions entre objets selon un point de vue temporel. L'accent est mis sur la chronologie des envois de messages. La **Figure II.3** représente un modèle de diagramme de séquence des collaborations entre objets selon un point de vue temporel.

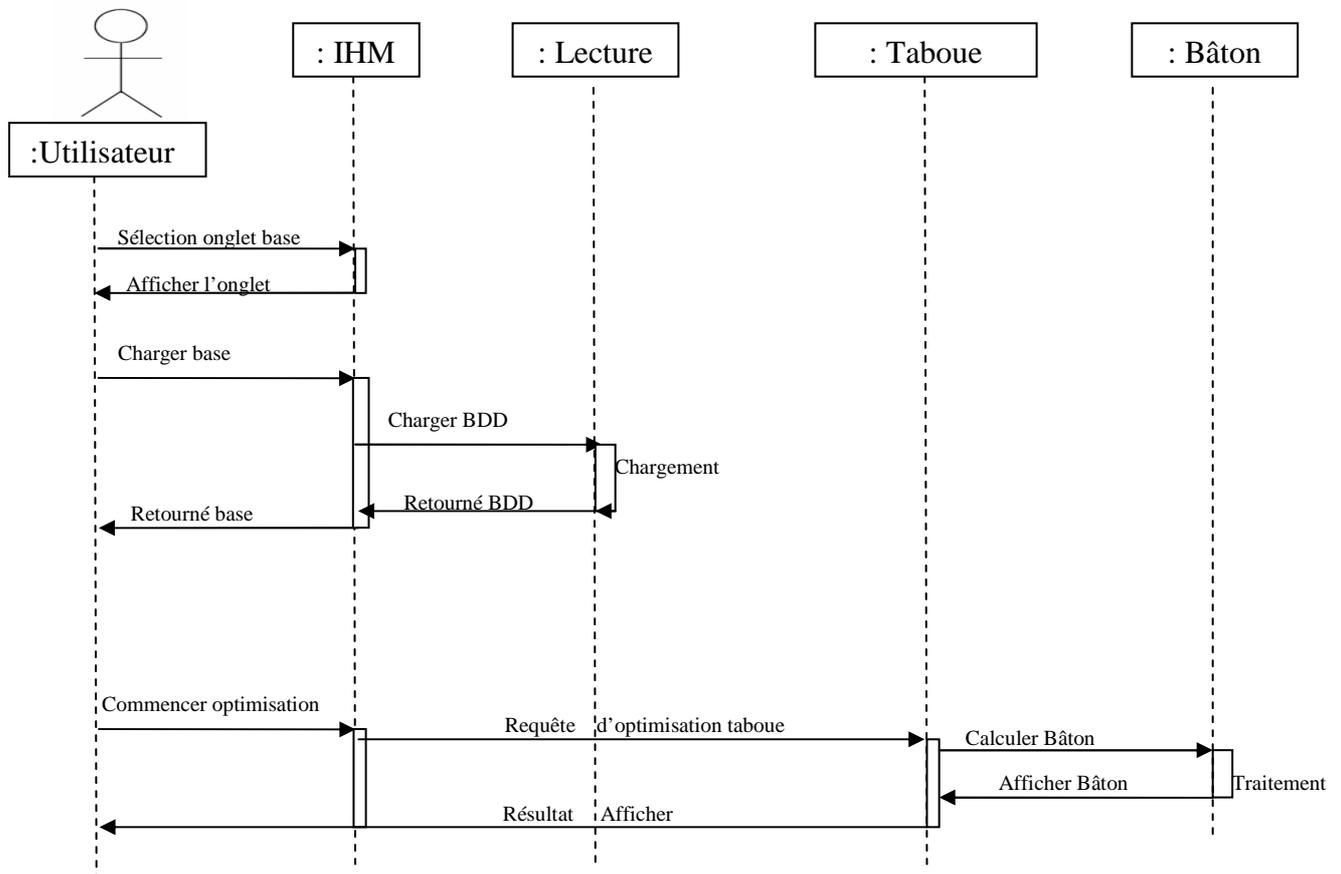
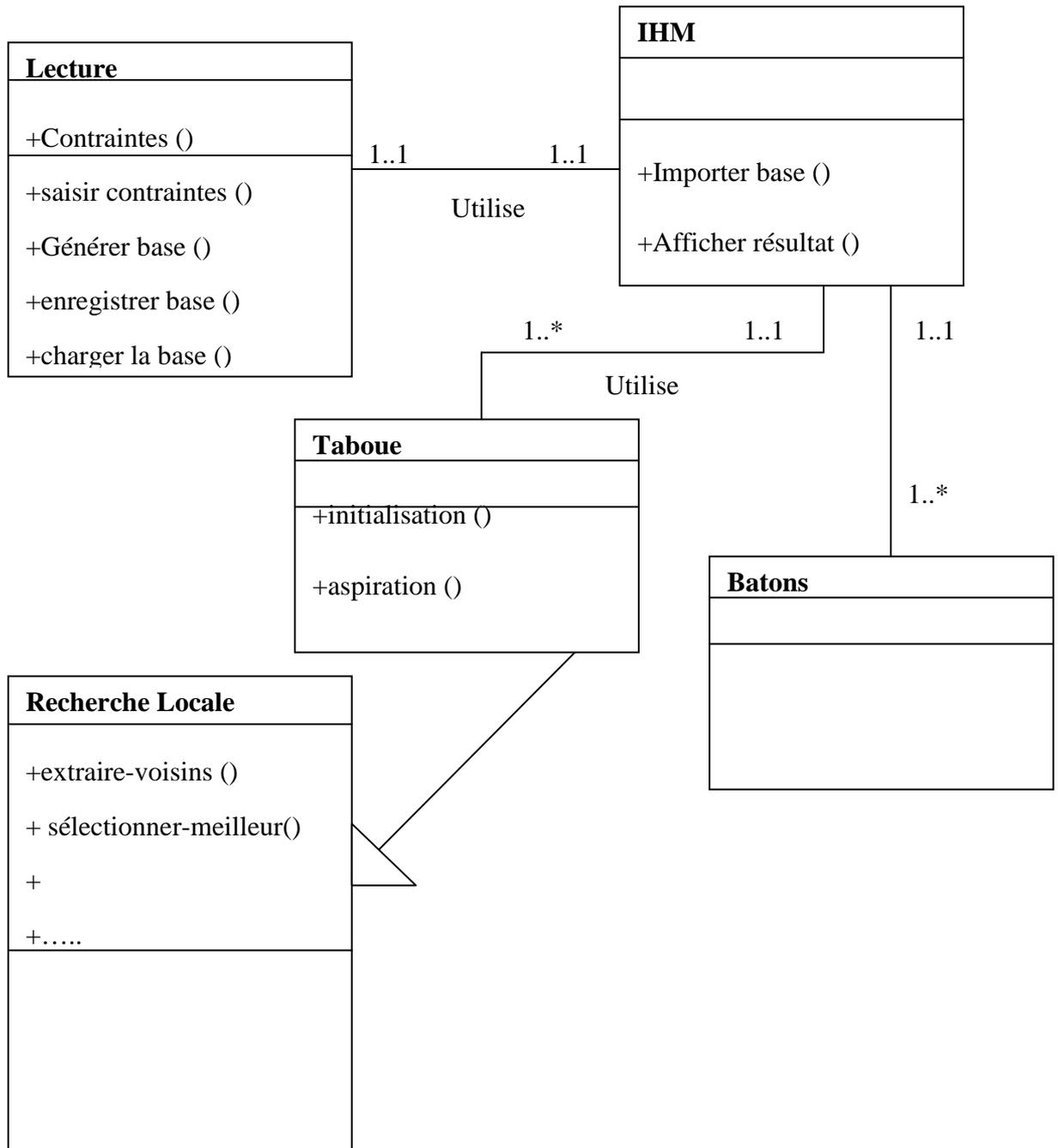


Figure II.3: Diagramme de séquence

### 3.3. Diagramme de classes

Un diagramme de classe est un diagramme le plus utilisé d'UML. Il décrit les types des objets qui composent un système et les différents types des relations statiques qui existent entre eux. La **Figure II.4** représente le diagramme de classes de notre application.



**Figure II.4: 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 fenêtre principale se compose de 03 onglets principaux :

- Le premier onglet : pour le chargement, génération, enregistrement et affichage de la base de données, le deuxième enregistre les contraintes globales de l'utilisateur, le troisième lance une recherche taboue.



Figure II.5: Interface principale.



Figure II.6: validation des contraintes.



Figure II.7: lancement de sélection.

## 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 7, Processeur I3, 4 Giga de RAM.

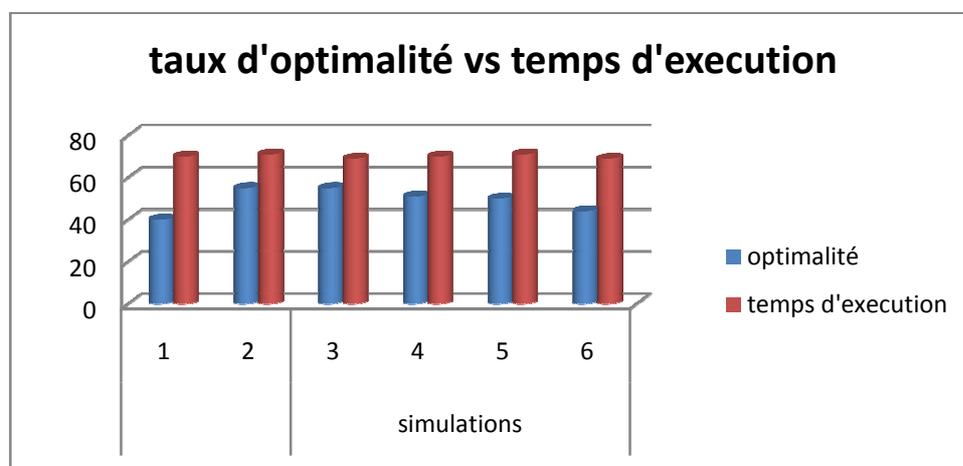
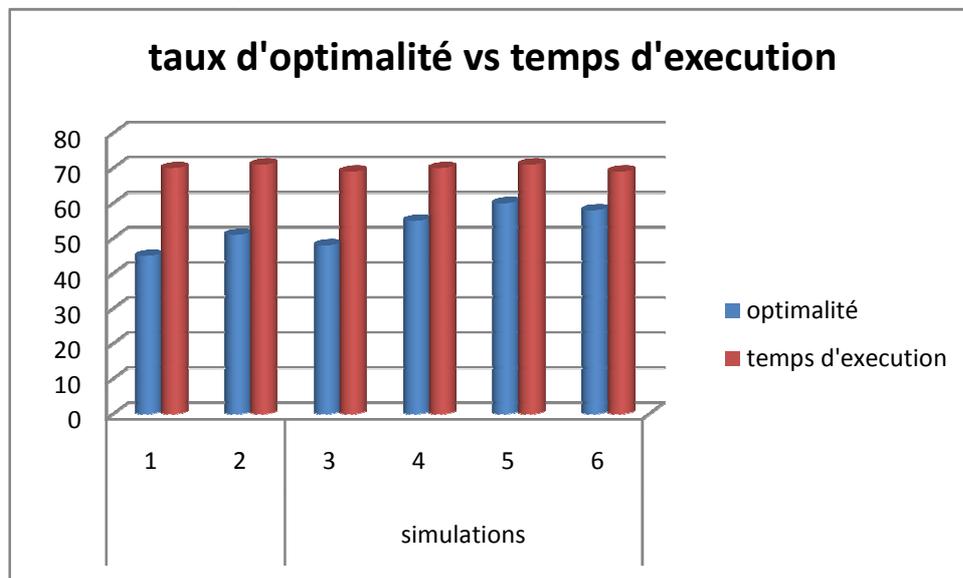


Figure II.8: histogramme d'optimalité pour la recherche taboue standard.



**Figure II.9: histogramme d'optimalité pour la recherche taboue améliorée**

▪ **Discussion**

On remarque que les résultats sont proche 55% pour la recherche taboue standard, et elle peut augmenter jusqu'à 60% d'optimalité lorsqu'on rajoute le redémarrage aléatoire,

**En effet**, nous avons redémarré l'algorithme si les fitness obtenus sont inférieures à 50% après 40% des itérations. Ces valeurs sont choisies de façon empirique, elles représentent les meilleures configurations.

En plus nous remarquons que le temps d'exécution est toujours le même, (environs de **70** seconds), parce qu'on alloué le même nombre d'itérations pour toutes les simulations (**1000** itérations).

## **6. Conclusion**

Nous avons présenté dans ce chapitre un algorithme d'optimisation mono objectif à base de recherche taboue. Nous avons montré aussi l'application développée ainsi que les expérimentations effectuées pour valider notre approche.



# 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 à la recherche taboue.

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.

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**

Pour le codage de notre projet, nous avons choisi le langage de programmation java et l'environnement de développement Netbeans.

❖ **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 Développment 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, refactoring, 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. [30]

✓ **Protocoles**

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

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

- ❖ **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.

## 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>
- [7] <http://www.siteduzero.com/informatique/tutoriels/les-services-web/qu-est-ce-qu-un-service-web>
- [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.

- [14] <http://www.ecr-france.org/boite-a-outils/glossaire-ecr/341-xml-extensible-markup-language-?lang=> .
- [15] Chauvet J., “Service Web avec SOAP, WSDL, UDDI, ebXML“; jouve, Paris; mars 2002 .
- [16] SoftDeath : Architecture d'un service Web 2003.
- [17] [www.journaldunet.com/developpeur/tutoriel/xml/030707xml\\_rest1a.shtml](http://www.journaldunet.com/developpeur/tutoriel/xml/030707xml_rest1a.shtml).
- [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/> .
- [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/> .
- [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’université JOSEPH FOURIER, 18 novembre 2008.
- [24] [http://fr.wikipedia.org/wiki/Service\\_Web#Avantages](http://fr.wikipedia.org/wiki/Service_Web#Avantages) .
- [25] David Chappell, et Tyler JEWELL, Java Web Service, édition O’Reilly, Mars 2002.
- [26] <http://www.w3.org/2001/XMLSchema-instance>
- [27] Glover F. , Future paths for integer programming and links to artificial intelligence, Computers and Operations Research, Vol. 13, pages 533-549, 1986
- [28] José Crispín HERNÁNDEZ HERNÁNDEZ, ALGORITHMES MÉTA-HEURISTIQUES HYBRIDES POUR LA SÉLECTION DE GÈNES ET LA CLASSIFICATION DE DONNÉES DE BIOPUCES, THÈSE DE DOCTORAT Spécialité : Informatique, 2008

- [29] 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.
- [30] Maurice Clerc : L'optimisation par essaim particulaire, Tutoriel pour OEP 2003.

## Résumé

Les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications sur l'Internet. Ils constituent la technologie de base pour le développement d'architectures orientées services. Ces architectures sont de plus en plus répandues sur le Web. Le principe essentiel du modèle de service Web est de transformer le Web en un dispositif distribué d'échange et de calcul, où les services Web peuvent interagir d'une manière intelligente.

Actuellement, de nombreux services Web, avec des fonctionnalités similaires sont fournis par des fournisseurs concurrents, un problème qui apparaît à la mise en œuvre de cette technologie est que le processus de sélection devient assez complexe.

Notre travail consiste à sélectionner, une composition de services à base de recherche Taboue .cette approche prend en compte les critères de QoS, pour instancier une composition abstraite.

**Mots Clés :** Services Web, Architectures Orientées Services, Sélection, recherche Taboue.

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 work an approach based on tabu search. 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, Quality of services, Optimization, Tabu Search.

ملخص

الكلمات الرئيسية: خدمات الويب، خوارزمية الأسراب، الخوارزميات الوراثية، جودة الخدمة، التحسين