



République Algérienne Démocratique et Populaire
Université Abou Bakr 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 Service Web à base d'algorithme Mimétique

Réalisé par :

- MOULAY KHATIR Oussama Mohy-eddine
- HENAOUI Redouane

Présenté le 02 juillet 2013 devant le jury composé de MM.

- Mr CHIKH Med Amine (Président)
- Mr HADJILA Fethallah (Encadreur)
- Mr BENZAOUZ Mortada (Examineur)
- Mr BELABED Amine (Examineur)

Année universitaire : 2012-2013

Remerciement

Louange à Dieu qui ma donné la force, le courage, et l'espoir nécessaire
Pour accomplir ce travail et surmonter l'ensemble des difficultés.

Je tiens tout d'abord à remercier mon encadreur **Mr. HADJILA Fethallah**
pour ces conseils et sa disponibilité et son encouragement qui nous ont
permis de réaliser ce travail dans les meilleures conditions.

J'exprime également ma reconnaissance à **Mr. CHIKH Med Amine**
D'avoir accepté de présider le jury.

Mes remerciements à **Mr BENAZZOUZ Mortada** Pour avoir bien voulu
juger ce travail

Mes remerciements s'adressent également à **Mr BELABED Amine** Pour
avoir bien voulu juger ce travail

Mes vifs remerciements A tous mes enseignants durant mon cursus
universitaire.

Je remercie aussi les personnes qui m'ont aidé et encouragé le long de ce
travail.

Dédicace

Je tiens à dédier ce travail à mes très chers parents, à la mémoire de mon père allah yerahmou qui m'a toujours encouragé dans mes études et à ma mère qui m'a donné un modèle de persévérance,

A mes chers sœur dévouées Nadjia et Douaa,

A mon frère Salih et ma belle-sœur Khadidja qui m'ont toujours soutenu,

A mes grands-mères et mes grands-pères a la mémoire du défunt,

A mon cher binôme sans qui ce travail n'aurait pas été réalisé

A tous les membres du département informatique,

A tous mes amis qui m'aime

Dédicace

Je tiens à dédier cet humble travail,

A mes très chers parents, pour leurs amour et sacrifices,

A mes chers frères,

A mes proches amis et toute ma grande famille, pour leurs soutient et
encouragements,

A toutes les personnes qui me connaissent de prés ou de loin

Résumé

La composition de services est l'une des motivations principales de l'architecture SOA, en effet l'utilisateur final à besoin de combiner plusieurs services pour satisfaire les requêtes complexes. Ce problème devient de plus en plus difficile, si on prend en charge la QOS pour les services web fonctionnellement équivalents.

Dans ce travail nous proposons une approche de sélection de services basée sur les algorithmes mimétiques. Les résultats expérimentaux montrent la supériorité de l'approche mimétique par rapport à l'approche génétique.

Mots clés : services Web, sélection de services web, algorithme mimétique, qualité de service, optimisation combinatoire, recherche locale.

Abstract

Recent progress in network and software engineering has resulted in the proliferation of web services. This technology allows to rapidly create new applications by reusing existing components.

Toward this end this work proposes a solution for the problem of QoS-aware service selection. The proposed approach is based on mimetic algorithms, this later combines genetic operators with local search.

The obtained results showed high efficiency of the suggested approaches.

Keywords : Web services, selection of web services, mimetic algorithm, quality of service, combinatorial optimization, local search.

ملخص

تعد عملية تركيب الخدمات من احد التحديات الأساسية للهيكلة الخدماتية. في الواقع قد يحتاج المستعمل الى تركيب عدة خدمات من اجل تلبية حاجياته, التي تكون معقدة في بعض الاحيان. و قد تزداد هذه الاشكالية صعوبة في حال اعتبار بعض مقاييس الجودة من اجل اختيار الخدمات, خاصة وان الويب يعرف منافسة كبيرة في هذا المجال, في هذا العمل نقترح مقارنة لتحسين انتقاء خدمات الويب وذلك بالاعتماد على خوارزمية "ميميستيك", وقد أظهرت نتائج العمل التطبيقي اهمية المقاربة خاصة بالمقارنة مع الخوارزميات الوراثة.

الكلمات المفتاحية : خدمات الويب, انتقاء خدمات الويب, خوارزمية "ميميستيك", نوعية الخدمات, مقارنة تحسينية, بحث محلي.

Table des matières

Liste des figures	3
Liste des tableaux	4
Introduction générale.....	5
Chapitre I Les Services Web.....	8
I. Introduction	9
II. Service web	9
II.1. Définitions.....	9
II.2. Architecture des Services Web	9
II.3. Les Standards des services web	11
a) XML, la langue maternelle des services web.....	11
b) SOAP, communiquer	11
c) WSDL	14
d) UDDI.....	15
II.4. Les avantages et inconvénient des services Web	16
III.Sélection des services web	17
III.1. exemple de motivation (la création d'une entreprise).....	17
III.2. critères de Qos (Quality of service).....	18
III.3. La fonction objectif (score ou fitness).....	20
III.4. Optimisation combinatoire	20
a) Définition	20
b) Approches d'optimisation combinatoire (approches de sélection)	21
IV. Conclusion.....	23
Chapitre II Conception et Implémentation du Prototype.....	24
I. Introduction	25
II. Algorithme Mimétique	26
II.1. Historique	26
II.2. Définition	26

II.3. Fonction de l’algorithme	27
II.4. Description de l’algorithme.....	28
II.5. Les domaines d’applications	29
II.6. Quelques avantages et inconvénients.....	29
III. Description de la base	29
III.1. Description de la base.....	29
III.2. Description de la requête	30
IV. Conception	31
IV.1. Diagramme de cas d’utilisation	31
IV.2. Diagramme de séquence.....	32
IV.3. Diagramme de classe	33
V. Interface Humain/Machine (IHM)	34
V.1. Fenêtre principale	34
V.2. Expérimentation.....	37
V.3. Résultats.....	37
V.3.1. Discussion	39
V.4 Comparaison génétique / mimétique	40
V.4.1 Discussion	41
VI. Conclusion	41
Conclusion générale	42
Bibliographie	43

Liste des figures

Figure I.1 : Architecture en Pile des services Web.....	10
Figure I.2 : Structure de message SOAP.....	12
Figure I.3 : Traitement d'un message SOAP.....	13
Figure I.4 : Structure et description WSDL.....	14
Figure I.5 : Structure de donnée de l'annuaire UDDI.....	16
Figure I.6 : exemple de motivation.....	18
Figure I.7 : Les approches de sélection.....	22
Figure II.1: Schéma des Algorithmes Mimétiques (MA).....	26
Figure II.2: Schéma de recherche local.....	28
Figure II.3: Diagramme de cas d'utilisation	31
Figure II.4: Diagramme de séquence.....	32
Figure II.5: Diagramme de classe.....	33
Figure II.6 : Histogramme d'optimalité et temps d'exécution des 5 simulations...39	
Figure II.7 : Histogramme de comparaison d'optimalité.....	40

Liste des tableaux

Tableau I.1 : Les domaines de valeurs de critères.....	19
Tableau II.1 : description de base de données.....	29
Tableau II.2 : 1^{er} simulation.....	37
Tableau II.3 : 2^{eme} simulation.....	38
Tableau II.4 : 3^{eme} simulation.....	38
Tableau II.5 : 5^{eme} simulation.....	38
Tableau II.6 : résultat des 3 simulations de comparaison.....	40

Introduction générale

1- Contexte :

Le web est devenu une source d'informations dynamique dans lequel l'information est produite sur demande. Dans ce cadre les services web sont devenus le facteur le plus attirant pour les entreprises actuelles.

Les services Web représentent un mécanisme de communication entre applications distantes à travers un réseau (internet ou intranet).

La présence d'un ensemble de services web similaires d'un point de vue fonctionnel (même comportement ou la même interface ie les entrées-sorties), poussent les utilisateurs à faire des choix à l'aide des critères de QOS, ces critères auront une grande importance lorsque le besoin des clients est complexe (ie des compositions).

2- Problématique :

L'objectif de ce travail est de sélectionner de façon optimale (ou proche de l'optimale) une combinaison de service web à base de leur qualité de services.

Cette combinaison doit maximiser les critères positifs (ex : Fiabilité) et minimiser les critères négatifs (ex : Temps d'exécution), en plus elle doit satisfaire les contraintes globales de l'utilisateur (ex : Cout < 200 euros).

3- Contribution :

Pour résoudre ce problème on propose dans ce mémoire une approche d'optimisation mono objective basée sur l'algorithme mimétique.

Les algorithmes mimétiques sont considérés comme l'un des méthodes les plus efficaces dans le domaine d'optimisation, ils ont montré leur faisabilité dans la résolution de plusieurs problèmes. Ils se basent sur l'hybridation de deux techniques les « Algorithmes Génétiques » et « les méthodes de recherche locale ».

4- Plan de travail :

Le mémoire est composé de deux chapitres, qui sont organisés comme suit:

Le premier chapitre

Nous avons présenté les concepts généraux des services web. Tout d'abord, nous commençons par les différentes architectures des services web. Ensuite, nous avons cité quelques avantages et inconvénients des services web. Après, nous avons présenté le problème de sélection avec un exemple de motivation. Enfin, on a montré un état de l'art qui regroupe les différentes approches proposées pour résoudre ce problème afin de conclure ce que nous avons fait.

Le deuxième chapitre

Nous avons présenté notre algorithme avec un petit historique, une définition, son fonctionnement, ces domaines d'application et enfin on a cité quelques avantages et inconvénients de l'algorithme mimétique.

La suite du chapitre décrit la conception et la réalisation de notre application.

Conclusion :

Ce projet de fin d'étude nous a permis d'instruire et améliorer nos connaissances.

Nous avons présenté dans ce mémoire les technologies liées aux services Web. Nous avons proposé aussi un algorithme de sélection qui se base sur l'algorithme mimétique pour l'optimisation des compositions.

Chapitre I

Les Services Web

I. Introduction

Depuis les années 70, la communication entre les entreprises ainsi avec leurs clients été très difficile en terme de gestion. Ceci a provoqué l'apparition de nouvelle technologie pour résoudre ce problème, quelque chercheurs ont exploité le web pour réaliser des applications afin de faciliter l'adaptation aux marché commerciale « application B2B : Business to Business » et « application B2C : Business to Consumer ».

Ces applications sont basées généralement sur l'architecture de SOA qui fournit un ensemble de principe pour permettre l'échange d'information.

II. Service web

II.1. Définitions

Définition 1 :

Le consortium W3C définit un service Web comme étant : « une application, ou un composant logiciel qui vérifie les propriétés suivantes » [1] :

- Il est identifié par un URI.
- Ses interfaces et ses liens peuvent être décrits en XML.
- Sa définition peut être découverte par d'autres services Web.
- Il peut interagir directement avec d'autres services Web à travers le langage XML en utilisant des protocoles Internet standards.

Définition 2 :

Selon notre point de vue un service web est une application accessible à travers l'internet. Elle est décrite par un langage standardisé. Elle sert a facilité la communication et l'échange d'information entre les utilisateurs/programmes à l'aide des protocoles internet.

II.2. Architecture des Services Web

L'architecture des services web est implémenté a l'aide des diverses technologies. Elle est organisé en quatre couches comme illustré sur la figure «I.1 » [2].

Chaque couche de la pile des services web répond à des fonctionnalités déférentes tel que la sécurité, la messagerie fiable, les transactions, le routage ect. Comme représenter sur la figure I.1 dans un souci d'interopérabilité les déférentes couche de la pile des

services web s'interfaçent avec des standards. Les fonctions de couches supérieures reposent sur celles de couches inférieures.

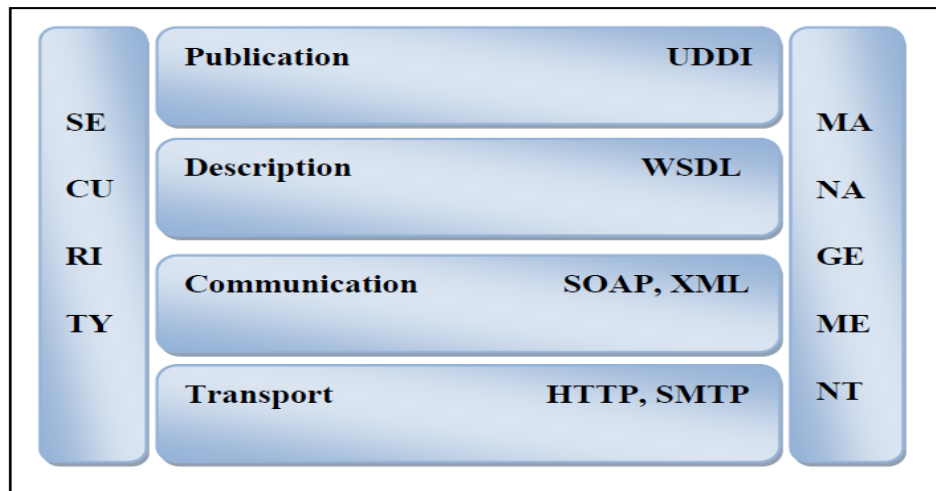


Figure I.1 : Architecture en Pile des services Web.

La partie du milieu de la figure représente les standards appliqués à chaque couche

- **La couche de publication:** repose sur le protocole UDDI (Universal Description, Discovery and Integration), qui assure le regroupement, le stockage et la diffusion des descriptions des services Web.
- **La couche description:** est prise en charge par le langage WSDL (Web Service Description Language) [Christensen et al., 2001], qui décrit les fonctionnalités fournies par le service Web, les messages reçus et envoyés pour chaque fonctionnalité, ainsi que le protocole utilisé pour la communication.
- **La couche communication:** la couche de communication des messages propose différents mécanismes liés à l'acheminement des messages (format de communication des messages, adressage, routage...etc.). Cette couche utilise des protocoles reposants sur le langage XML, car sa syntaxe unique résout les conflits syntaxiques lors de l'encodage des données. Actuellement, SOAP (Simple Object Access Protocol) est le protocole le plus utilisé pour cette couche.
- **La couche transport:** le protocole le plus utilisé dans cette couche est l'HTTP (Hyper Text Transfert Protocol). Cependant, d'autres protocoles peuvent être utilisés, tels que le SMTP (Simple Mail Transfer Protocol) ou le FTP (File

Transfer Protocol), permettant ainsi aux services Web de rester indépendants du mode de transport utilisé.

II.3. Les Standards des services web

a) XML, la langue maternelle des services web

XML [3] est un standard promulgué par le W3C, l'organisme chargé de standardiser les évolutions de web. On retrouve dans XML une généralisation des idées contenues dans HTML et SGML.

XML permet de définir des balises extensible et indépendante de l'aspect graphique que doit revêtir la page dans le navigateur web. Dans XML les balises permettent d'associer toutes sortes d'informations au fil du texte.

Il 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 documents.
- Séparation entre structure des documents et présentation des documents [4].

b) SOAP, communiquer

SOAP[5] est une spécification de communication entre services web, par échange de messages en XML au travers du web. Simple et facile à implémenter dans les serveurs web ou dans les serveurs d'application, SOAP est indépendant des langages de programmation ou des systèmes d'exploitation employé pour l'implémentation des services web.

SOAP est un protocole de communication entre applications fondé sur XML, visant à satisfaire un double objectif : servir de protocole de communication sur les intranets, dans une optique d'intégration d'applications d'entreprise, et permettre la communication entre applications et services web, en particulier dans un contexte d'échanges inter-entreprises (le réseau internet) [4].

Structure du message SOAP :

Les messages échangés lors de l'utilisation du protocole SOAP sont basés sur le langage XML. Ils sont composés de deux parties, l'en-tête de protocole de transport et l'enveloppe SOAP (la Figure I.2) [6].

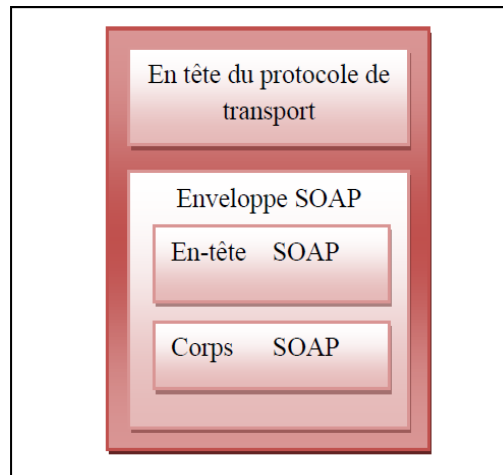


Figure I.2 : Structure de message SOAP

a) 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).

b) 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 du message 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 nœuds intermédiaires, lors de traitement du message.

Un nœud 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).

Les principaux attributs des éléments qui forment le bloc d'en-tête sont :

env:role: permet d'indiquer à quel nœud la fonction décrite est destinée.

env:mustUnderstand: c'est une valeur booléenne, elle permet de préciser que le traitement devient obligatoire pour un nœud intermédiaire, par exemple, pour un calcul très long, il peut être utile d'envoyer un e-mail à chaque étape.

env:relay: cet attribut permet de relayer un message à un autre nœud si le premier nœud n'est pas capable de le traiter.

- Le corps du message SOAP: l'élément corps de message SOAP contient des données spécifiques à l'application. 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).

Fonctionnement :

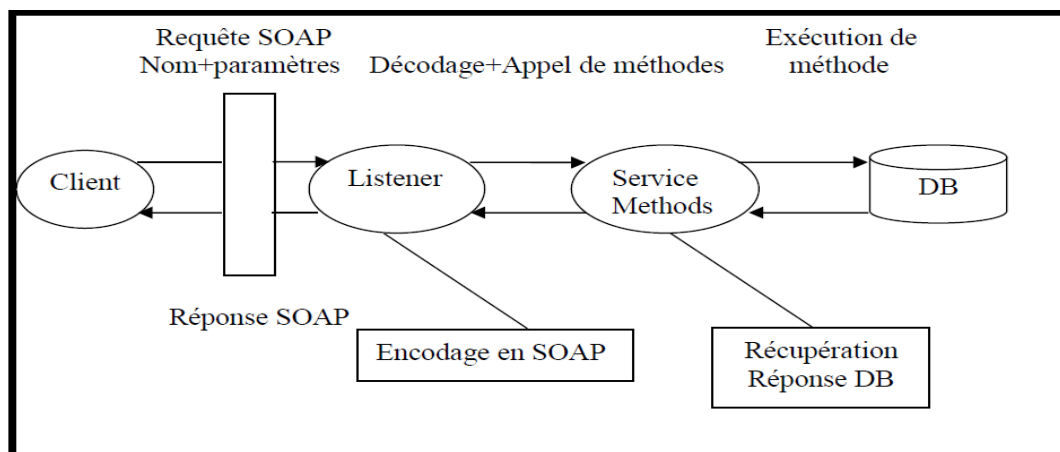


Figure I.3 : Traitement d'un message SOAP [6]

La requête SOAP transmise par le client a l'aide des protocoles de transmissions (exemple HTTP).

Si l'exécution des méthodes a besoin de des informations de la base de données alors il doit connecter avec la base de données. [7]

c) WSDL

WSDL[8] a été développé conjointement par IBM, Microsoft et Arriba et a été présenté pour analyse au W3C qui l'a accepté comme une notice et publié sur leur site.

L'utilité de WSDL est de décrire et publier le format et les protocoles d'un Web service de manière homogène par l'utilisation du format XML. Cela permettra au requérant et à l'émetteur d'un service de comprendre les données qui seront échangées

Il est simplement un IDL (Interface Definition Language) tout comme Corba IDL ou les interfaces de Java ou de COM par exemple[9].

Structure et description WSDL :

WSDL est divisé en trois éléments majeurs pouvant être séparés et utilisés indépendamment ou être combinés pour former un seul document XML. Ces éléments se subdivisent en 7 types de composantes descriptives dans la définition des services réseaux. Ces éléments sont (figure I.4) [10]

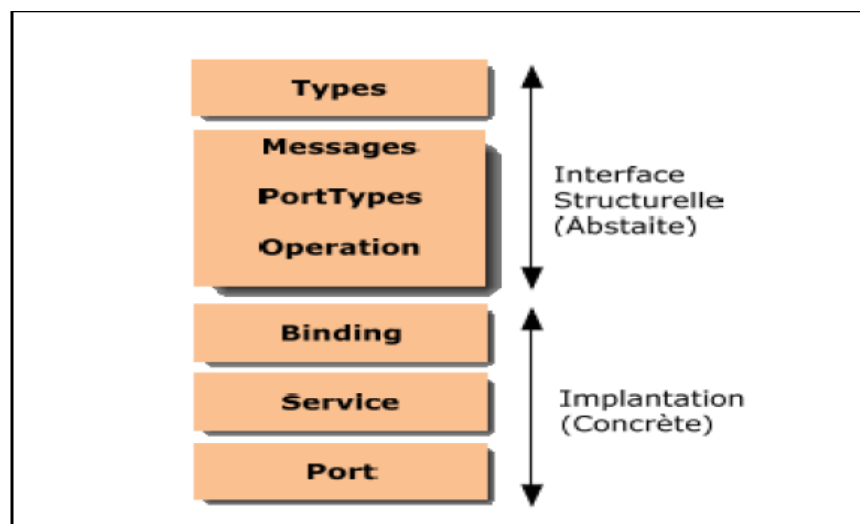


Figure I.4 : Structure et description WSDL

- **Data Type Définition** Identification du contenu et du type de données qui sont dans le message.

Data types- une enveloppe pour les « data type définitions » utilisant des systèmes

Tel que XSD (XML Schéma Définition)

Message- une définition abstraite du type de données qui est communiqué.

- **Abstract Operations** Définition de la manière dont les données seront échangées.

Operation- La description abstraite d'une action supportée par le service.

Port Types- Un ensemble d'opérations supporté par un ou plusieurs point d'accès (port).

Binding- Un protocole spécifique et une spécification du format de données pour un point d'accès particulier (port).

- **Service Binding** Définition de la couche transport qui servira au message.

Port- Un point d'accès unique définit comme une combinaison de l'adresse Réseau et du point d'accès particulier (port).

Service- Un ensemble de terminaisons reliées.

d) UDDI

Principe :

Le standard UDDI [11] défini par l'OASIS vise à décrire une manière standard de publier et d'interroger les services web proposés par un réseau, généralement au sein d'un service d'annuaire recensant les services web de l'organisation [12].

UDDI est un modèle d'information composé de structures de données persistantes appelées entités. Ces entités doivent être décrites en XML, et stockées dans les nœuds UDDI [13].

Structure de donnée de l'annuaire UDDI :

Selon « Arnaud » [14], Un registre UDDI se compose de quatre types de structures de données : le businessEntity, le businessService, le bindingTemplate et la tModel (figure I.5) [14].

Cette répartition par type fournit des partitions simples pour faciliter la localisation rapide et la compréhension des différentes informations qui constituent un enregistrement.

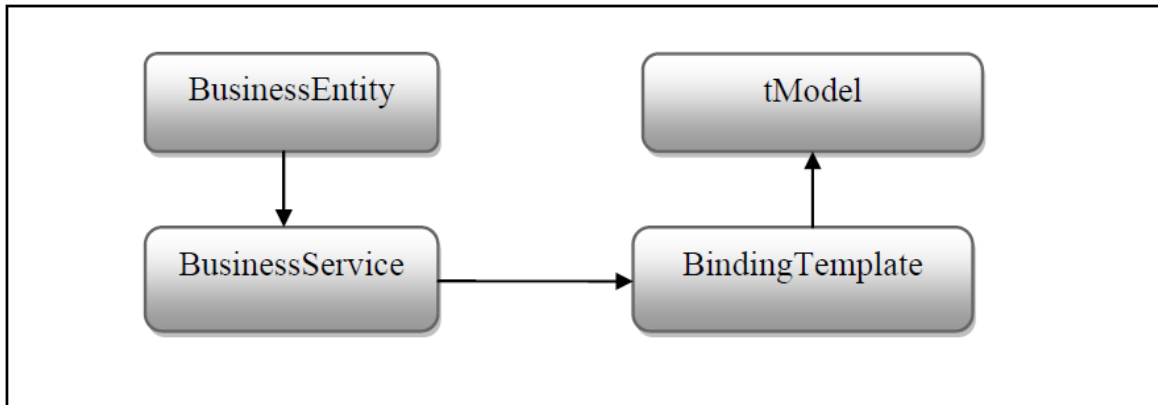


Figure I.5 : Structure de donnée de l'annuaire UDDI

- **BusinessEntity** : contiennent des informations concernant les entreprises fournisseurs de services, la recherche se fait par contact, nom et adresse de fournisseur.
- **BusinessService** : présentent les services selon leurs fonctionnalités en suivant une taxonomie industrielle standard, le service est recherché par sujet et par domaine.
- **BindingTemplate** : offrent des informations techniques sur les services web ainsi que la référence de la description WSDL. Le service est recherché en fonction de ses caractéristiques techniques.
- **Tmodel** : ensemble d'information sur le mode d'accès au service (définitions wsdl), il peut s'agir aussi d'une spécification abstraite ou d'une taxonomie [15].

II.4. Les avantages et inconvénient des services Web

- Avantages

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les services Web utilisent des standards et protocoles ouverts.

- 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-feu 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. [16]

- **Inconvénients**

- Les normes de services Web dans certains domaines sont actuellement récentes.
- 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. [17]

III. Sélection des services web

III.1. exemple de motivation (la création d'une entreprise)

Dans cette partie, nous [18] présentons un scénario dont l'objectif est d'illustrer et de motiver notre approche pour la sélection de services. Ce scénario concerne la création d'une entreprise, et pour cela nous avons besoins de 3 services (figure I.6) [18] :

1/ un service qui fournit les informations nécessaires pour la création d'une entreprise.

2/ un service qui fournit des informations sur les concurrents du marché de même type d'activité.

3/ un service qui fournit des allocateurs pour l'hébergement de cette entreprise.

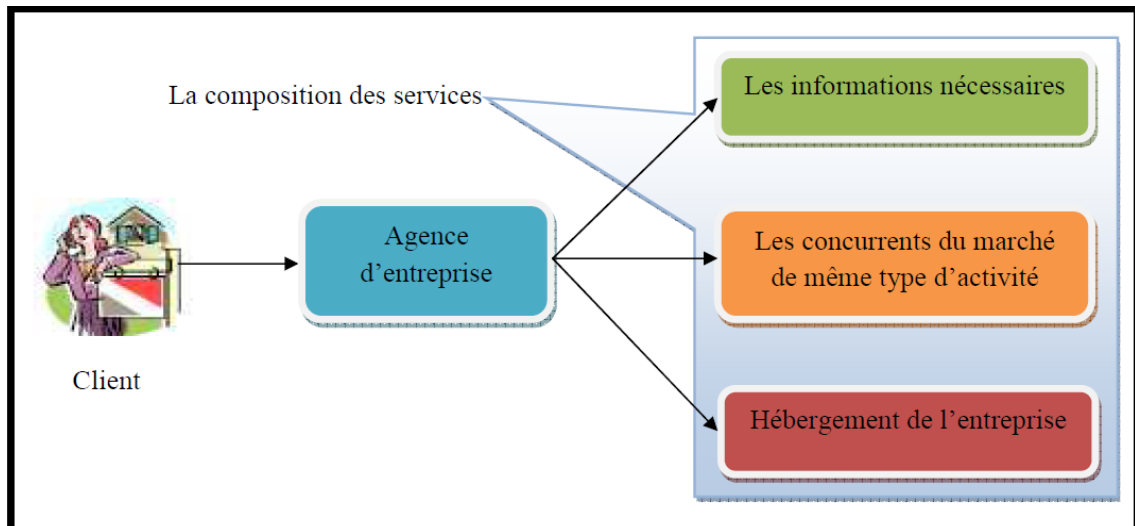


Figure I.6 : exemple de motivation

Chaque service est présenté par un ensemble de compagnie de telle sorte à respecter un ensemble de critères de qualité. Pour chaque critère, nous fournissons une définition, indiquons sa granularité, et fournissons des règles pour calculer sa valeur pour un service donné.

III.2. critères de Qos (Quality of service)

Cinq critères de qualité génériques pour des services élémentaires sont considérés :

- ✓ **Latence** : Latence mesure le retard prévu en secondes entre le moment où l'exécution commence et moment où l'exécution finit. La latence est calculé par :

$$Latence(sol) = \sum_{i=1}^n lat(C_i)$$

- ✓ **Cout** : le prix d'exécution est la somme d'argent qu'un demandeur de service doit payer pour exécuter une opération d'un service. Les allocataires de service annoncent directement le prix d'exécution de leurs opérations.

$$Cout(sol) = \sum_{i=1}^n cout(C_i)$$

- ✓ **Réputation** : la réputation d'un service est une mesure de sa fidélité. Elle dépend principalement des expériences de l'utilisateur d'employer ce service.

Les différents utilisateurs peuvent avoir le service à peu près identique de différents avis. La valeur de la réputation est définie comme rang moyen indiqué au service par des utilisateurs.

$$Rep(sol) = \frac{1}{n} \sum_{i=1}^n rep(Ci)$$

- ✓ **Disponibilité :** La disponibilité d'un service est la probabilité que le service est accessible.

$$Disp(sol) = \log_2 \prod_{i=1}^n disp(Ci)$$

- ✓ **Fiabilité :** La fiabilité d'un service est la probabilité qu'une demande est correctement répondue dans le délai de temps prévu maximum (ce critère calcul le taux d'erreur). La fiabilité est une mesure liée à la configuration de matériel et/ou de logiciel des services et aux connexions réseau entre les demandeurs de service et les fournisseurs.

$$Fiab(sol) = \log_2 \prod_{i=1}^n fiab(Ci)$$

On suppose que les domaines de valeurs de critères de Qos sont définis comme suit :

Critères	Domaine de valeur
cout	1...30euros
Temp d'exécution	1...300ms
fiabilité	0...5
disponibilité	0...1
réputation	0...1

Tableau I.1 : Les domaines de valeurs de critères

III.3. La fonction objectif (score ou fitness)

Ces critères doivent optimiser une fonction globale qui maximise les critères positifs (Réputation, Disponibilité et Fiabilité) et minimise les critères négatifs (cout et temps d'exécution), et nous devons également satisfaire des contraintes globales par exemple :

- Cout (sol) < 200euros : Le cout d'une solution doit être moins de 200euros.
- Temps (sol) < 1000s : Le temps d'exécution doit être moins de 1000ms.
- Rep (sol) > 3 : La réputation doit être supérieure à 3.
- Disp (sol) > 0.6 : La disponibilité doit être supérieure à 60%.
- Fiab (sol) > 0.6 : La fiabilité doit être supérieure à 60%.

Nous définissons une fonction score qui nous facilite la comparaison entre les qualités de chaque composition de web services :

$Q_i(\text{sol}) = (\text{Cout}(\text{sol}), \text{Rep}(\text{sol}), \text{Disp}(\text{sol}), \text{Fiab}(\text{sol}), \text{Temps}(\text{sol}))$

$$U(C) = \sum_{Q_i \in \text{Neg}} W_i \frac{Q_i \max - Q_i \text{sol}}{Q_i \max - Q_i \min} + \sum_{Q_i \in \text{Pos}} W_i \frac{Q_i \text{sol} - Q_i \min}{Q_i \max - Q_i \min}$$

Où $W_i \in R_0^+$ et $\sum_{i=1}^n W_i = 1$ est la somme des poids assignés pour chaque critère de qualité pour représenter les priorités de l'utilisateur.

Les contraintes globales doivent être vérifiées par la condition suivante :

$$Q(k(C)) \geq \text{Cont}_i$$

Et pour intégrer les contraintes globales exigées par l'utilisateur on a calculé un nouveau score de pénalité donné par la formule suivante :

$$P(C) = \sum_{i=1}^5 (\text{Cont}_i - Q_i)^2$$

Donc le score final de la fonction objective sera le suivant :

$$F(C) = U(C) - P(C)$$

III.4. Optimisation combinatoire

a) Définition

L'optimisation est une branche des mathématiques qui permet de résoudre des problèmes en déterminant le meilleur élément d'un ensemble selon certains critères

prédéfinis. De ce fait, l'optimisation est omniprésente dans tous les domaines et évolue sans cesse depuis Euclide.

Un problème d'optimisation en général est défini par un espace de recherche S et une fonction objective f . le but est de trouver la solution $s^* \in S$ de meilleure qualité $f(s^*)$. Suivant le problème posé, on cherche soit le minimum, soit le maximum de la fonction f [19].

b) Approches d'optimisation combinatoire (approches de sélection)

Plusieurs efforts de recherches ont été rapportés dans la littérature sur la sélection automatique de service [19]. Les travaux existants sont scindés en deux types [20] :

1. L'optimisation multi-objective : appelée aussi l'optimisation a base de skyline (the skyline based optimization) peut être manipulé en employant les méta-heuristiques ou les techniques de base de données [21]. Plus spécifiquement nous pouvons employer l'algorithme de clivage et conquérir, l'algorithme bitmap, l'algorithme basé par index (B tree, Hash table), et l'algorithme de plus proche voisin (R tree).

D'ailleurs, il y a plusieurs travaux qui tient compte des préférences d'utilisateur pour choisir les K top horizons dominants [20 ; 21] certains d'entre eux utilisent la théorie des ensembles flous pour modeler les préférences et le rapport de dominance, les autres emploie le concept de Pareto-dominance pour ranger les services de web.

2. L'optimisation mono-objective : comporte plusieurs approches, elle peut employer un modèle de sélection global [22 ; 23 ; 24] ou un modèle de sélection local [25] ou un modèle de sélection hybride [19].

- Le modèle de sélection global se concentre sur toute la composition (ie les n composants de la solution), il peut obtenir la solution optimale, mais il a une complexité exponentielle.
- Néanmoins, le modèle local a seulement une complexité linéaire mais ne peut pas manipuler les contraintes globales (il manipule seulement des contraintes locales).
- Le troisième modèle est un compromis des deux approches, il commence la recherche par une optimisation globale, puis il continue le travail avec

une optimisation local, sa complexité temporelle est inferieur par rapport à l'optimisation globale, et il peut également manipuler les contraintes globales.

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

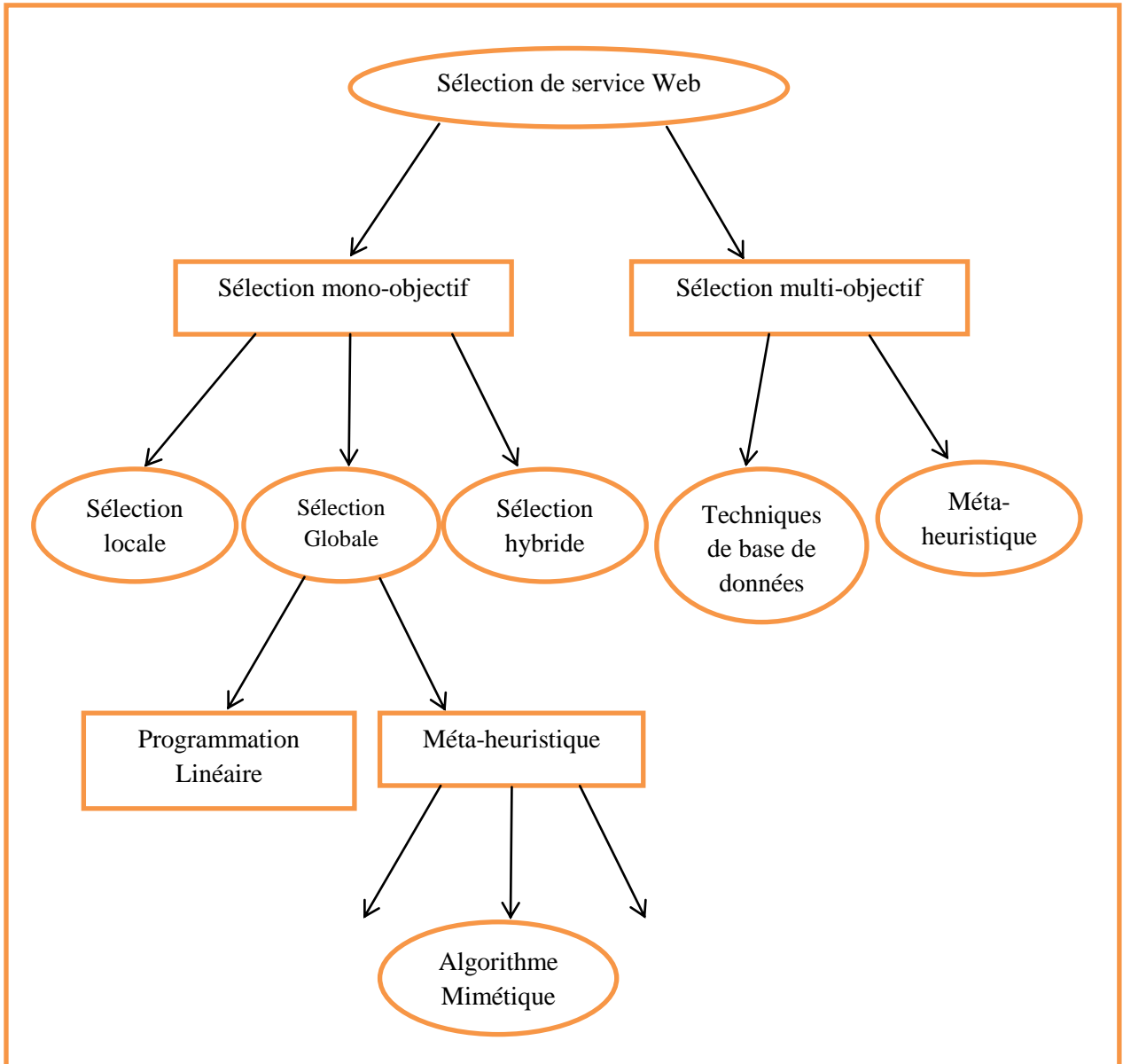


Figure I.7 : Les approches de sélection

IV. Conclusion

Dans ce chapitre on a présenté quelques définitions sur les services web avec une architecture ainsi les standards de communication entre les services web.

Après on a présenté le problème de sélection avec un exemple de motivation, en suite on a montré un état de l'art qui regroupe les différents approches proposé pour résoudre ce problème.

Dans le chapitre suivant nous proposons une méta-heuristique basée sur la sélection à partir de l'algorithme mimétique afin de rechercher la meilleure composition qui satisfait l'ensemble des contraintes globales.

Chapitre II

Conception et Implémentation du Prototype

I. Introduction

Les requêtes orientées services permettant aux utilisateurs d'accéder à plusieurs services Web de manière transparente et efficace. De façon générale, des fournisseurs concurrents peuvent offrir des services Web similaires fonctionnellement mais différents d'un point de vue QOS. un défi majeur est de concevoir des stratégies d'optimisation pour trouver les meilleurs services Web ou de la composition de celle-ci à l'égard de la qualité attendue fournie par l'utilisateur (au niveau par exemple de temps, de coût, et de réputation), Les normes existantes à base de technologies de découverte de services sont nettement insuffisants pour la construction d'une infrastructure de service à part entière de requête, le paradigme actuel de recherche par mots clés ne peut pas toujours localiser précisément les services Web en partie à cause de la sémantique riche énoncés dans ces services. [21]

Dans ce chapitre nous définissons notre algorithme de sélection et sa méthode de fonctionnement.

Le noyau de ce chapitre est d'avoir connaître la fonction de notre projet à partir d'une interface facile à manipuler qui aide à sélectionner le meilleur service pour l'utilisateur.

II. Algorithme Mimétique

II.1. Historique

Les algorithmes mimétiques sont une hybridation entre les algorithmes de recherche locale et les algorithmes génétiques. Les principes de ce type d'algorithme ont été introduits par Dawkins et formalisés par Moscato (Dawkins 89, Moscato 89, Moscato 99). Ils sont appelés aussi algorithmes génétiques hybrides, et Recherche locale Hybrides [26].

II.2. Définition

Les algorithmes mimétiques [27 ;30] sont une hybridation entre les algorithmes de recherche locale et les algorithmes génétiques. Le principe général est le même que pour les algorithmes génétiques mis à part qu'un opérateur de recherche locale est ajouté après celui de mutation. La partie génétique de ces algorithmes peut être vue comme une forte diversification alors que la partie recherche locale correspondrait à une forte intensification (accompagnée d'une faible diversification).

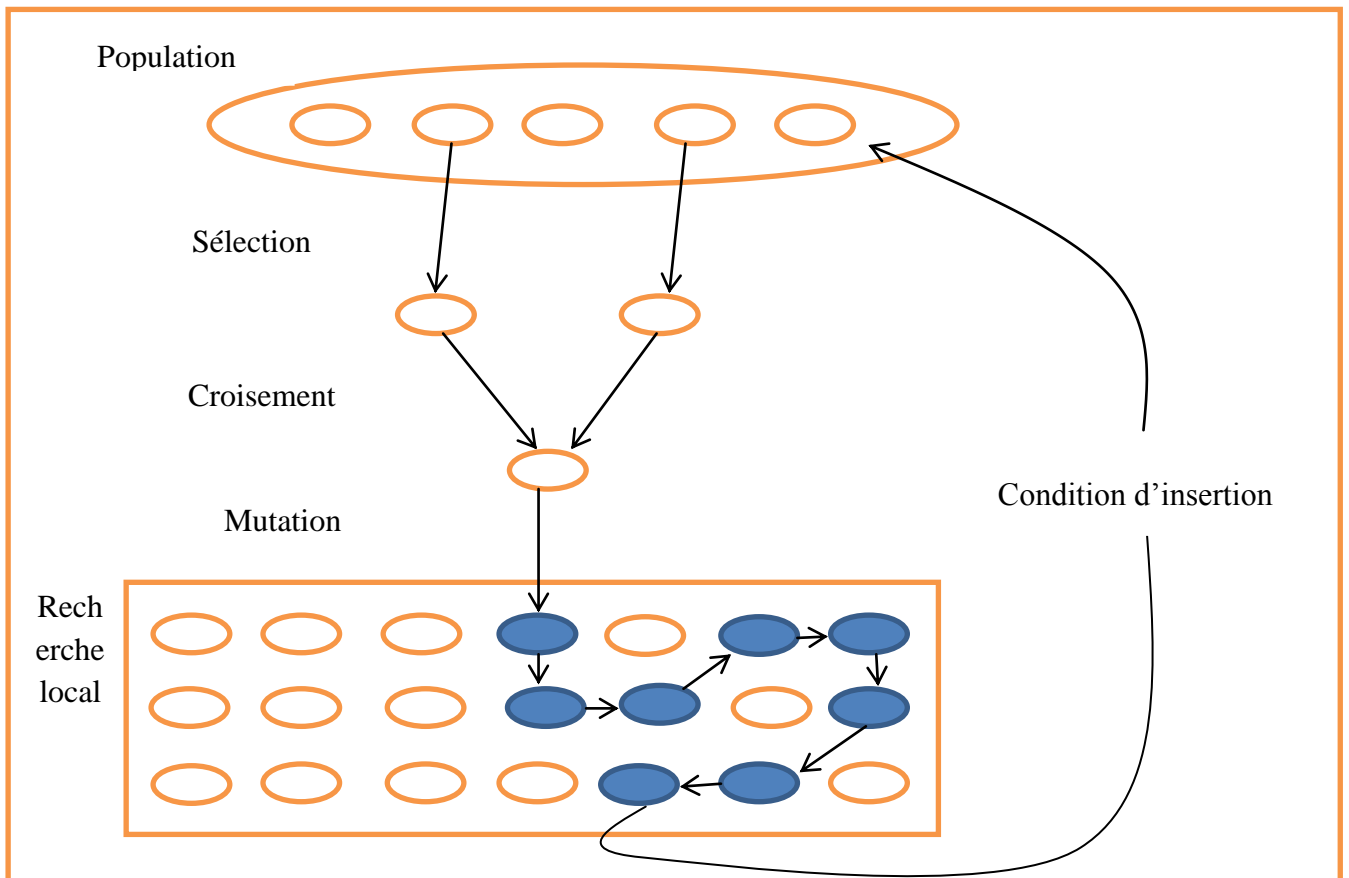


Figure II. 2: Schéma des Algorithmes Mimétiques (MA)

Si un individu vérifie cette condition, il peut être inséré dans la population. Dans le cas contraire, il est détruit et une nouvelle génération est construite. Cette condition est très importante car elle définit la politique d'évolution de la population.

Les algorithmes mimétiques sont des métaheuristiques avancées ; l'idée principale de cette technique est de rendre un algorithme génétique plus efficace par l'ajout d'une recherche locale en plus de la mutation. Une des observations générales provenant de l'implémentation d'un algorithme génétique basique est souvent la faible vitesse de convergence de l'algorithme. L'idée de Moscato est donc d'ajouter une recherche locale qui peut être une méthode de descente ou une recherche locale plus évoluée (recuit simulé ou recherche tabou). Il est évident que cette simple modification entraîne de profonds changements dans le comportement de l'algorithme.

II.3. Fonction de l'algorithme

- 1: Initialisation : générer la population initiale Pop de solutions avec taille = n
- 2: Améliorer chaque solution s de Pop : $s \leftarrow RL(s)$
- 3: Répéter
- 4: Sélectionner deux solution x et x' avec la technique de sélection
- 5: Croiser les deux parents x et $x' \rightarrow$ enfants $C1, C2$
- 6: Muter C
- 7: Pour chaque enfant C faire
- 8: Améliorer :si $rand() \leq proba-RC$ alors $C \leftarrow RL(C)$
- 9: Remplacer une solution P de Pop par C
- 10: Fin pour
- 11: Jusqu'à (critère d'arrêt).

L'intensification est produite par l'application de la recherche locale et l'opérateur de mutation assure la diversification [26].

La figure II.2 [27] explique le principe de la recherche locale

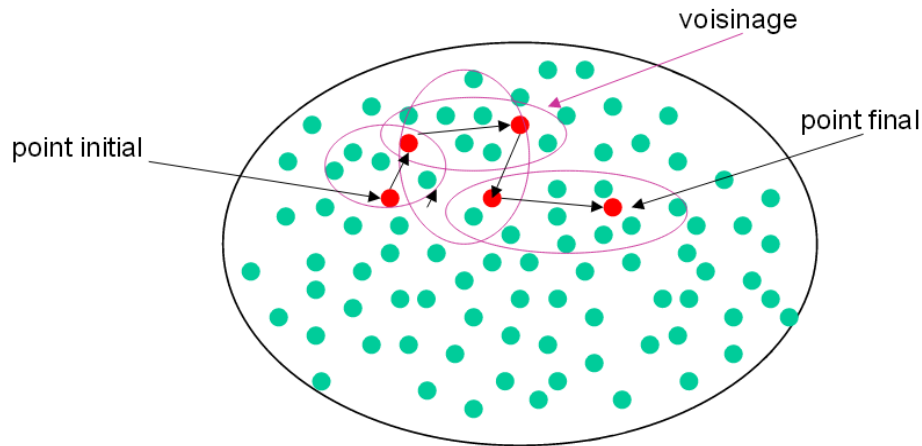


Figure II. 2: Schéma de recherche local

II.4. Description de l'algorithme

- **Etape 1 :** construire une population initiale de $|pop|$ individus de manière aléatoire ou avec une initialisation gloutonne.
- **Etape 2 :** appliquer l'opérateur de sélection Le choix de ces individus peut se faire de différentes manières. Il est souhaitable que les parents possèdent de bonnes propriétés afin de les transmettre aux fils.
- **Etape 3 :** appliquer l'opérateur du croisement Un opérateur de croisement permet de créer un nouvel individu à partir de deux individus parents.
- **Etape 4 :** appliquer l'opérateur de mutation est utilisé pour introduire de nouveaux gènes dans la population. (Pour quitter l'optimum local).
- **Etape 5 :** appliquer l'opérateur de recherche locale aux nouveaux individus créés (enfants). Si la probabilité actuelle est inférieure à la probabilité fixée auparavant (prob-RC) Pendant n itérations et renvoie les meilleurs individus trouvés (Population des minima locaux).
- **Etape 6 :** mise à jour remplacer P par C (si la fitness de c est supérieure à celle de P).

Le critère d'arrêt est différent d'un problème à un autre.

Il est évident que cette modification (l'ajout de la recherche locale) va changer le comportement de l'algorithme :

- La première utilisation de la procédure recherche locale permet de travailler avec des solutions bien intensifiées et qui sont parmi les meilleures.
- La deuxième utilisation de la recherche locale permet d'obtenir les meilleures enfants de la génération nouvellement créée.

II.5. Les domaines d'applications

Les algorithmes mimétiques [28] ont montré leur utilité pour la résolution de plusieurs problèmes:

- La reconnaissance des formes.
- Problème de Voyageur de Commerce (PVC)
- Problème Coloration de Graphes.
- Problèmes d'ordonnement.

II.6. Quelque avantages et inconvénients

- + L'ajout d'une recherche locale à un algorithme génétique peut compenser la faiblesse des AG en vitesse de convergence qui est très lente.
- + Sans la gestion efficace d'une population de solutions, il est difficile pour une recherche locale de parcourir efficacement l'espace des solutions souvent très vaste [28].
- L'ajout d'une étape de recherche locale provoque une augmentation du temps d'exécution de l'algorithme original

III. Description de la base

III.1. Description de la base

Nous avons créé un schéma de service qui contient dix classes de services web. Le nombre d'instances dans chaque classe est de quarante (40) fournisseurs de services web. Chaque fournisseur se caractérise par des qualités de service. Nous utilisons cinq paramètres pour évaluer les opérations des services : latence, fiabilité, disponibilité, cout et réputation. Les valeurs de ces paramètres sont générées en se basant sur une distribution uniforme. Un échantillon de notre base est illustré dans la table III.1 qui représente le fournisseur *i* des cinq premières classes.

critere	intervale
cout	1 - 30
Temp d'execution	1 - 300
fiabilité	0.5 - 1
disponibilité	0.7 - 1
réputation	0 - 5

Tableau II.1 : description de base de données [21]

Nous avons multiplié les valeurs des critères de couts et de latence par -1, pour homogénéiser la fonction à optimiser ie : nous voulons maximiser tous les critères positifs (fiabilité, disponibilité, réputation) et négatifs (cout, Temp d'execution)

III.2. Description de la requête

La requête de l'utilisateur comporte cinq éléments :

- La borne minimale pour la fiabilité.
- La borne minimale pour la disponibilité.
- La borne minimale pour la réputation.
- La borne maximale pour le cout.
- La borne maximale pour latence.

IV. Conception

IV.1. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Il permet aussi d'identifier les possibilités d'interactions entre le système et les acteurs.

La figure (II.3) représente le diagramme de cas d'utilisation de notre application.

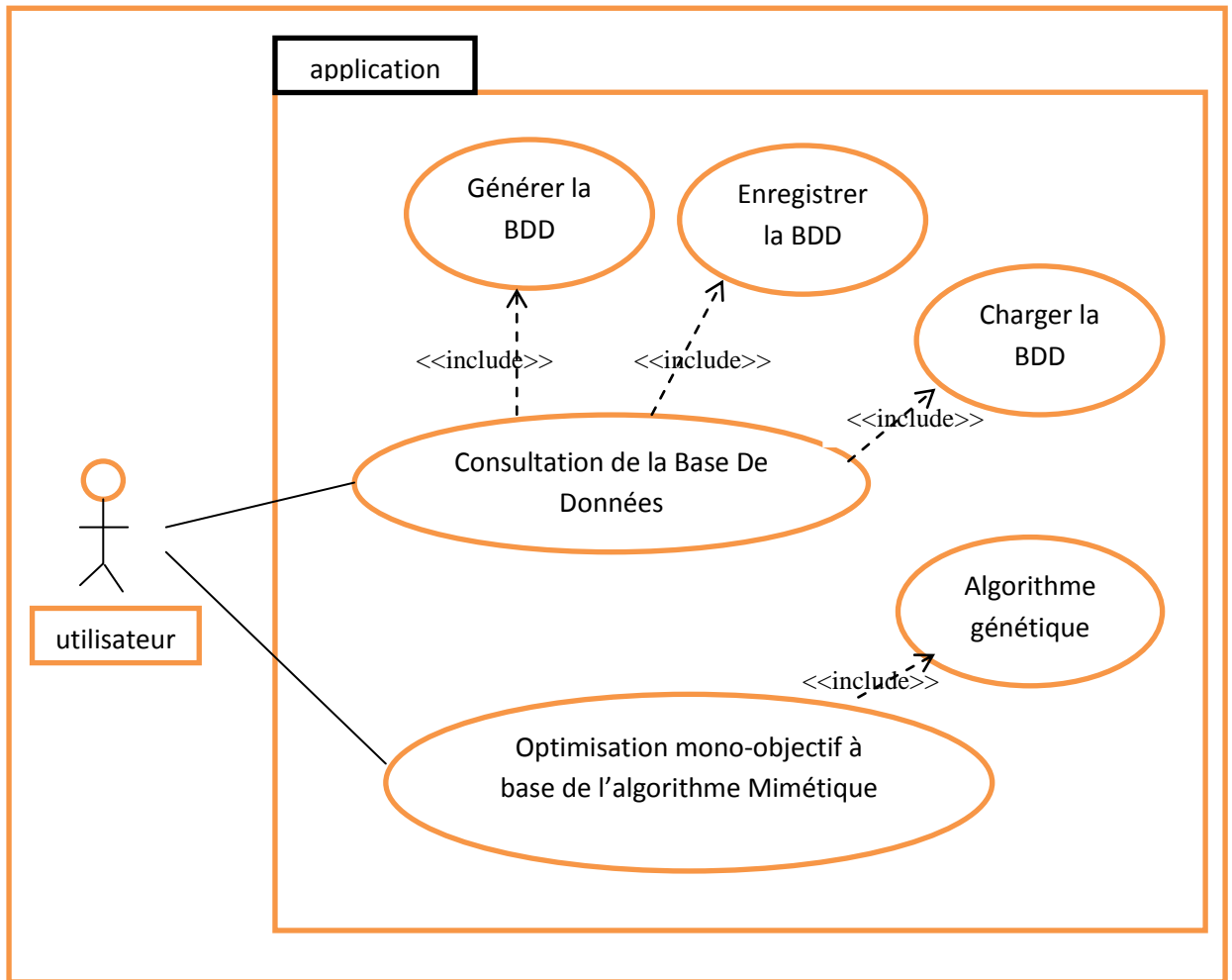


Figure II.3: Diagramme de cas d'utilisation

IV.2. Diagramme de séquence

Un diagramme de séquence permet de représenter des collaborations entre objets selon un point de vue temporel. Les objets communiquent en échangeant des messages représentés sous forme de flèches. Il peut servir à illustrer un cas d'utilisation. Un exemple de diagramme de séquence de notre application est représenté dans la figure II.4

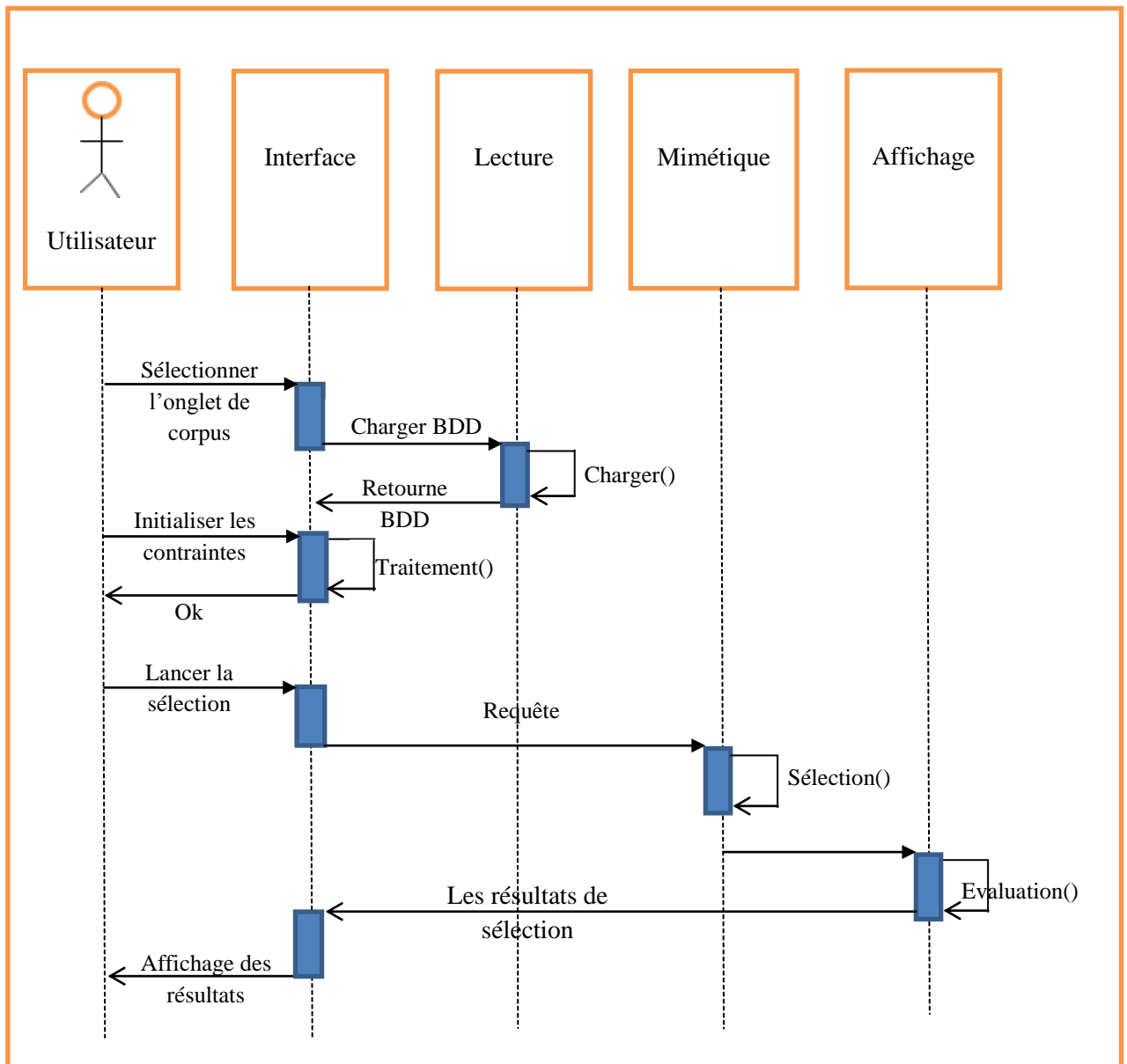


Figure II.4: Diagramme de séquence

IV.3. Diagramme de classe

Un diagramme de classes permet de représenter les classes intervenant dans le système. Il constitue un élément très important de la modélisation et permettre de définir quelles seront les composantes du système final. Il permet de structurer le travail de développement de manière très efficace. La figure II.5 représente le diagramme de classes de notre application.

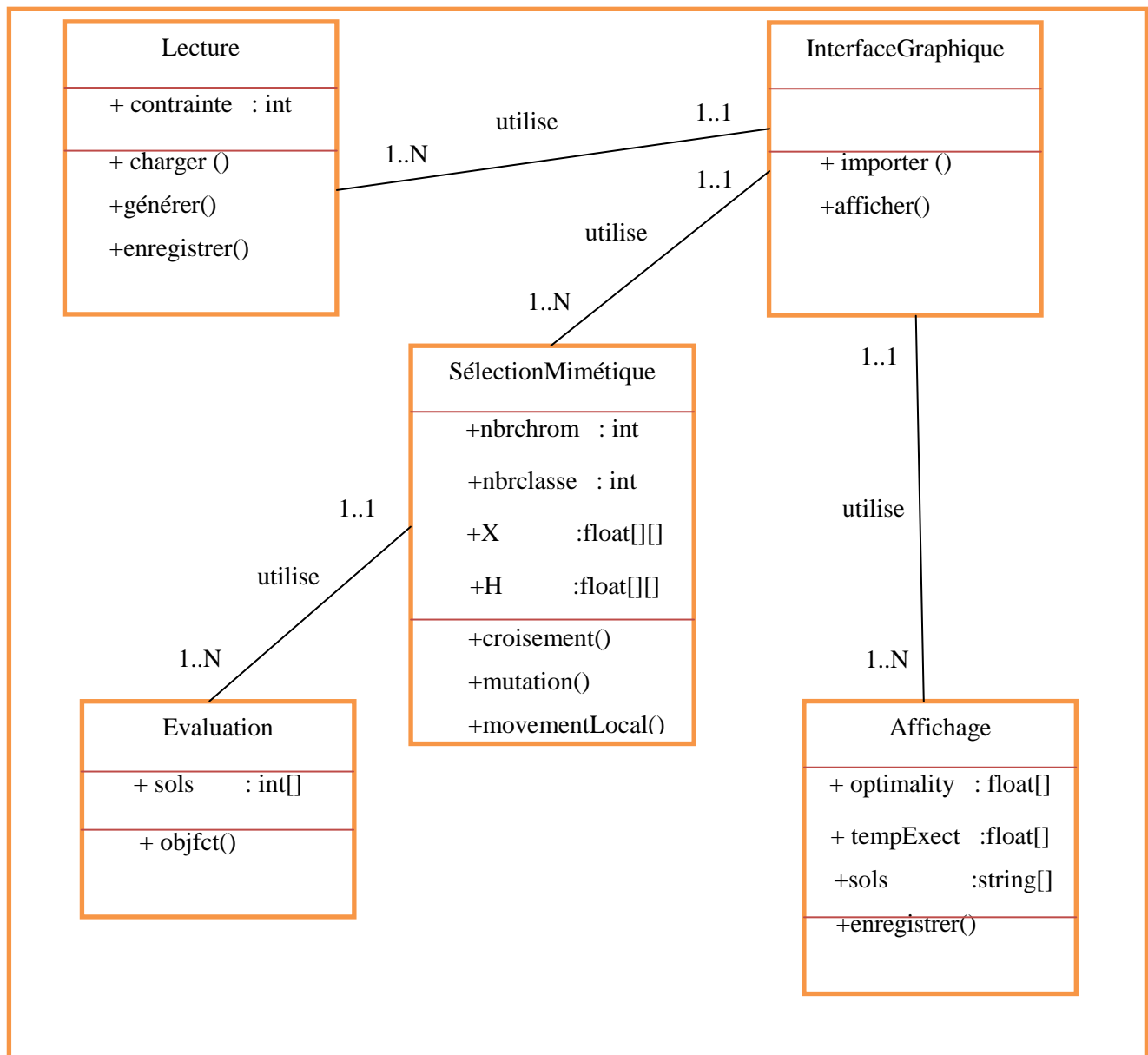


Figure II.5: Diagramme de classe

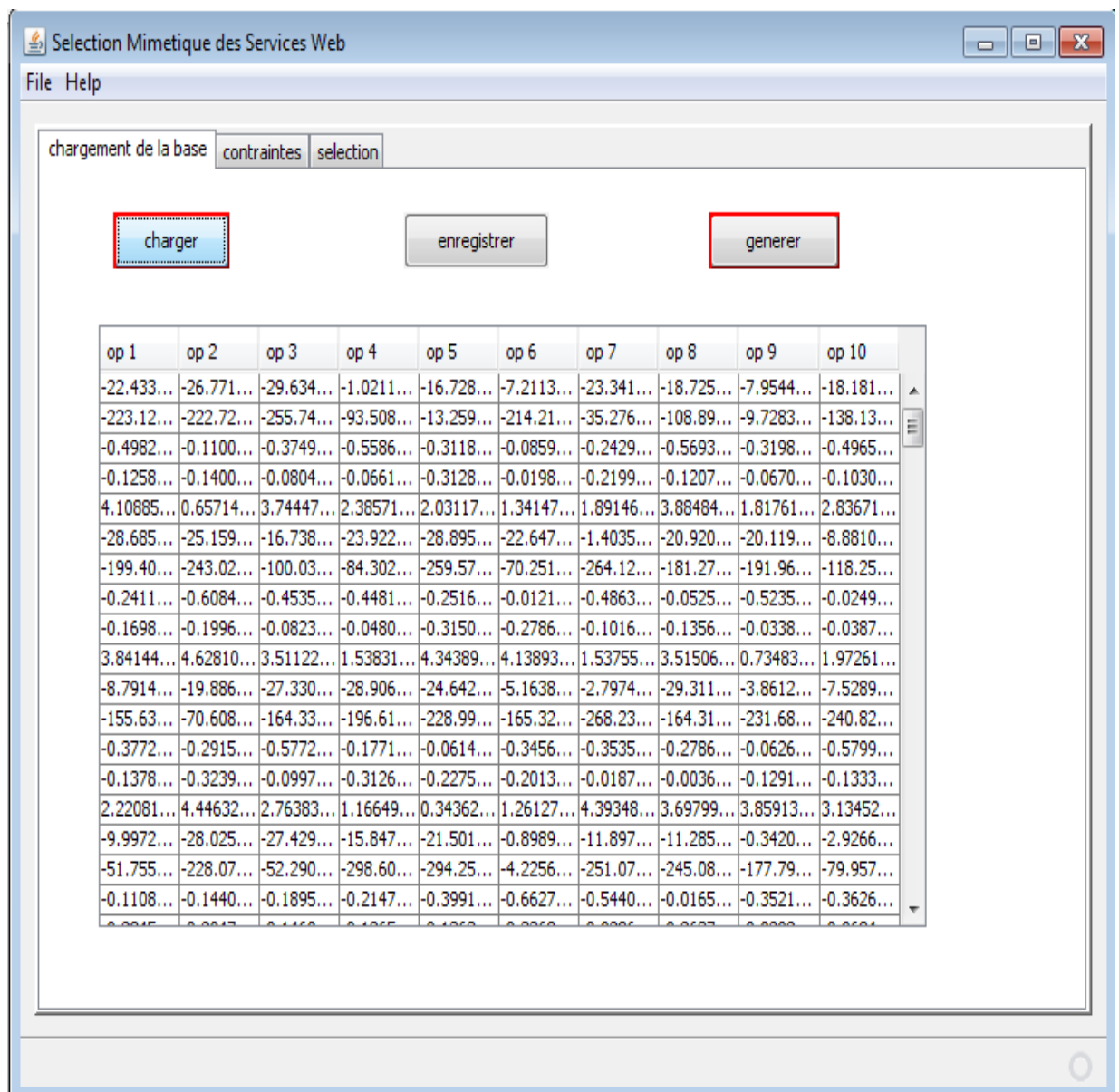
V. Interface Humain/Machine (IHM)

L’IHM 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 faciles à manipuler.

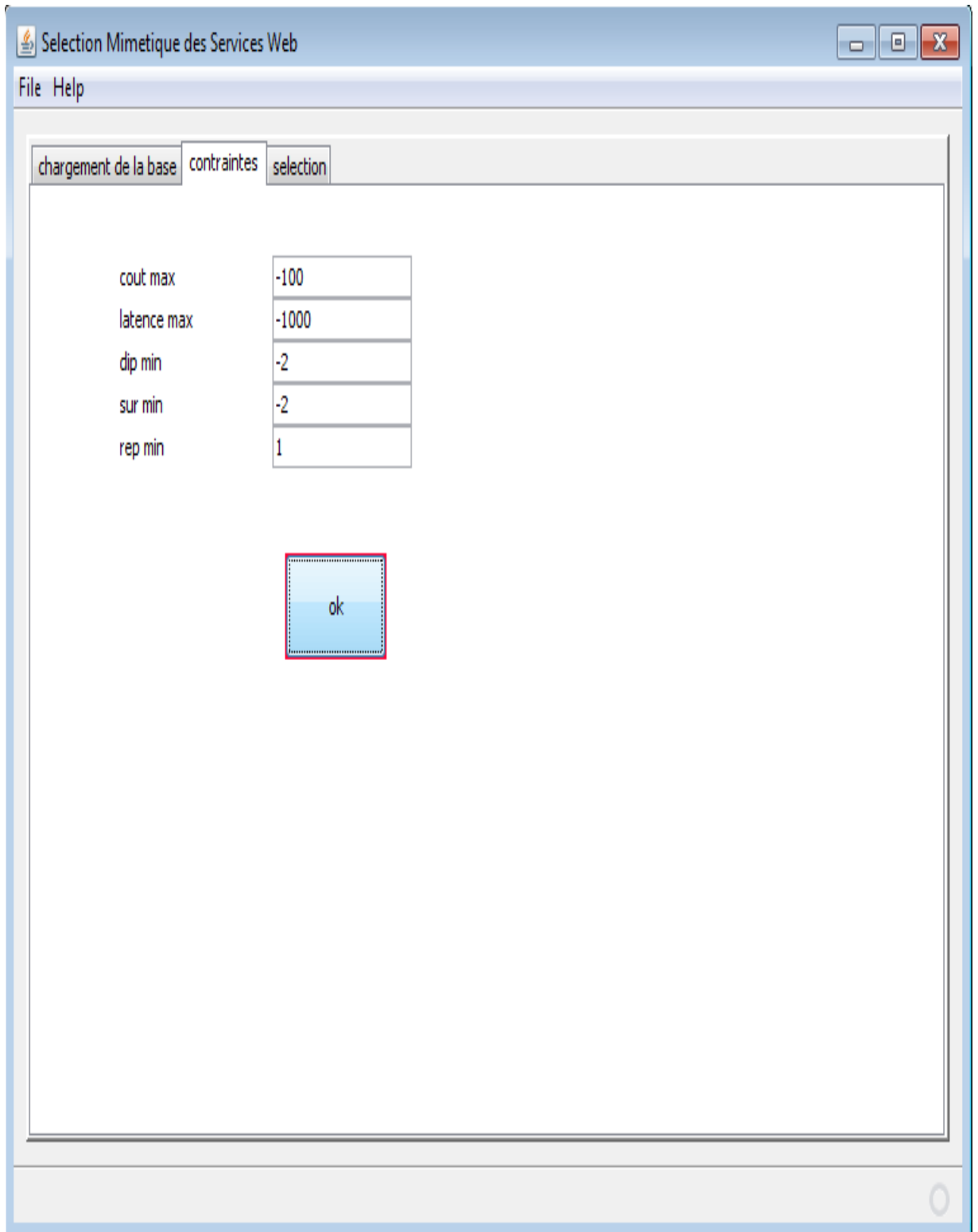
V.1. Fenêtre principale

La fenêtre principale se compose de trois onglets :

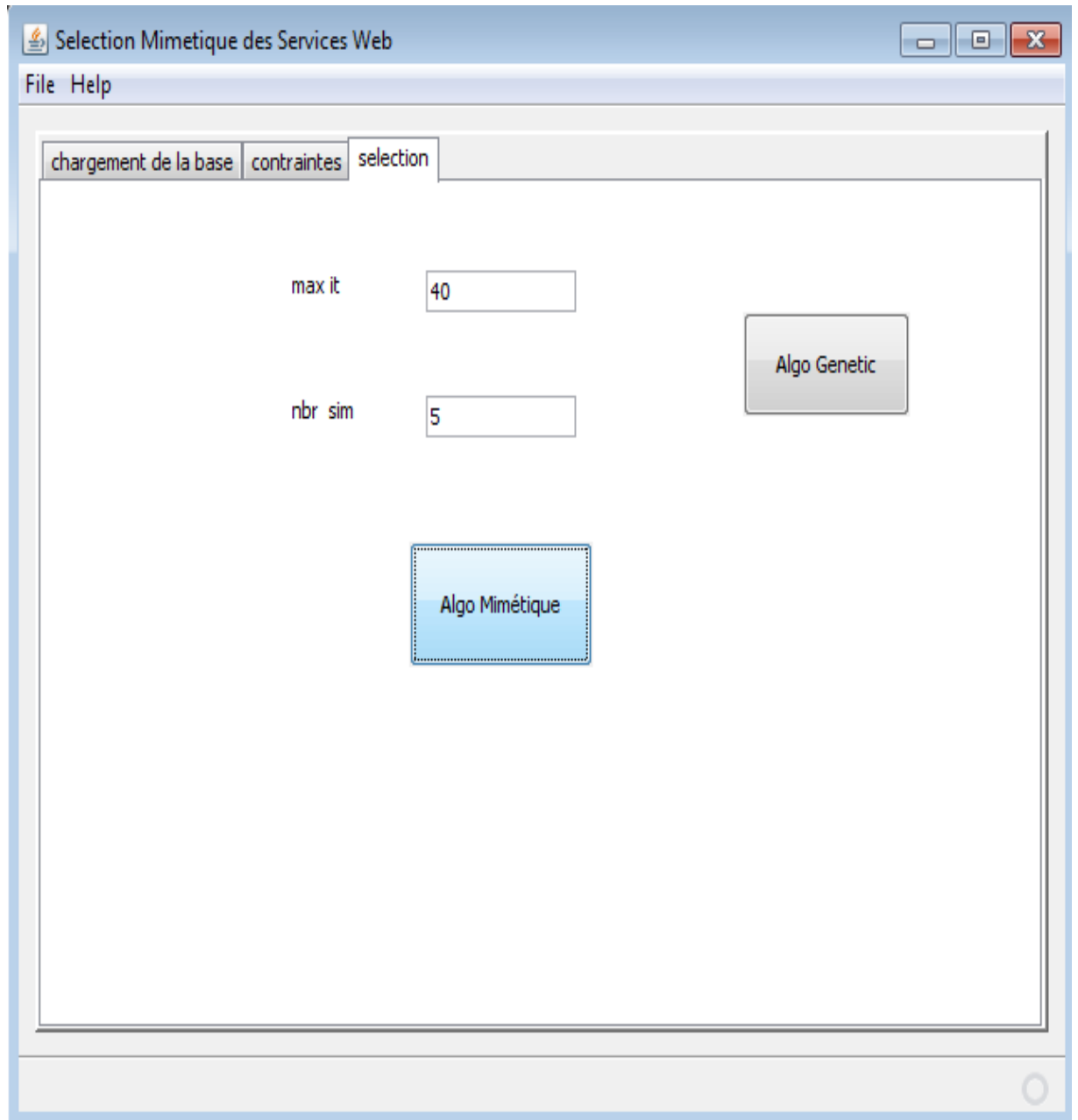
- Le premier onglet : pour le chargement, génération, enregistrement et l’affichage de la base de données.



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



- Le troisième onglet : ce dernier onglet est destiné pour la sélection à base d'Algorithme Mimétique, dont l'utilisateur a le choix de saisir le nombre d'itération et simulation qui lui convient.



V.2. Expérimentation

Nous avons mené une expérience pour évaluer la performance de l'approche proposée. Nous utilisons le scénario précédant (exemple de motivation du chapitre I) comme notre environnement de test pour configurer les paramètres de l'expérience. Le but est de démontrer comment notre approche peut aider le client à sélectionner la meilleure offre. Nous avons développé notre prototype sous NetBeans IDE 6.8 de Sun Microsystems sous le système d'exploitation Windows, Processeur intel Core i3, 4 Go de RAM.

La figure (II.6) montre 5 simulations de l'algorithme à base de sélection mimétique.

V.3. Résultats

Les résultats des fitness ainsi que leurs optimalité et le temps d'exécution de toute simulation sont présentés dans les tableaux suivants, pour cela nous gardons le même nombre d'itérations et de simulations :

quelques fitness obtenues	Score fitness	optimalité	Temps d'exécution (sec)
1	0.19723883	0.2509399	
2	0.3350586	0.4262832	
3	0.59338504	0.7549427	
4	0.6656437	0.8468749	
5	0.6203469	0.7892454	
6	0.6656437	0.8468749	
7	0.67125666	0.85401616	
8	0.65888584	0.8382771	
9	0.65888584	0.8382771	
10	0.5689632	0.7238717	
11	0.3253286	0.41390407	
12	0.5106575	0.6496914	
Max	0.67125666	0.85401616	33

Tableau II.2 : 1^{er} simulation

Num fitness et max	Score fitness	optimalité	Temps d'exécution (sec)
1	0.25003794	0.3181144	
2	0.4459483	0.5673642	
3	0.067204975	0.0855025	
4	0.17941554	0.228264	
5	0.48880923	0.6218946	
6	0.2651062	0.3372852	
7	0.08920659	0.11349438	
Max	0.48880923	0.6218946	65

Tableau II.3 : 2^{ème} simulation

Num fitness et max	Score fitness	optimalité	Temps d'exécution (sec)
1	0.12637304	0.1607799	
Max	0.12637304	0.1607799	95

Tableau II.4 : 3^{ème} simulation

4^{ème} simulation :

Fitness nul (Pas de solution pour cette simulation)

Temps d'exécution de 4 simulations est 119 secondes.

Num fitness et max	Score fitness	optimalité	Temps d'exécution (sec)
1	0.24776547	0.3152232	
2	0.17066239	0.2171277	
3	0.4146987	0.5276064	
4	0.53802514	0.6857103	
Max	0.53802514	0.6857103	145

Tableau II.5 : 5^{ème} simulation

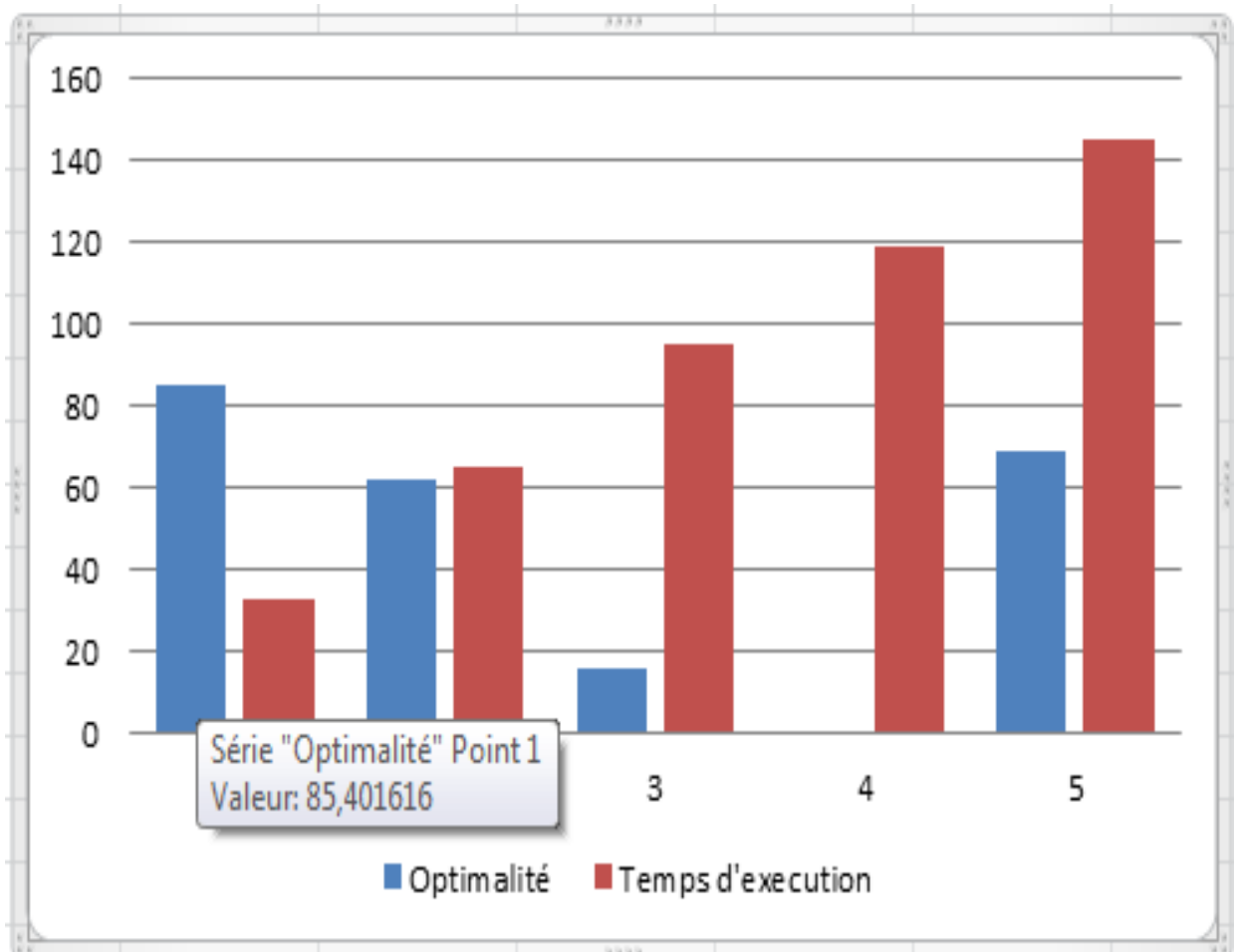


Figure II.6 : Histogramme d'optimalité et temps d'exécution des 5 simulations

V.3.1. Discussion

D'après nos expériences on conclut les points suivants :

- Plus le nombre d'itérations est élevé mieux seront les résultats d'optimalité malgré l'augmentation du temps d'exécution.
- Plus le pourcentage de croisement est élevé mieux seront les résultats, jusqu'à ce qu'on arrive à 80% (pour notre algorithme on' a fixé le croisement à 80% dès le début).
- La Même chose pour le pourcentage de mutation ie après plusieurs simulations nous avons conclu que la valeur 10% est la meilleure configuration pour le taux de mutation.
- Apres plusieurs modifications de la probabilité de recherche locale, nous avons conclu que la valeur 0.5 est la meilleure configuration

V.4 Comparaison génétique / mimétique

Le but de cette comparaison est d'étudier l'efficacité de l'algorithme mimétique par rapport à l'algorithme génétique.

La figure (II.3) montre les résultats de cette comparaison pour 3 simulations à chacune et chaque simulation à 100 itérations.

	Algorithme génétique		Algorithme mimétique	
	fitness	optimalité	fitness	optimalité
1^{er} sim	0.0	0.0	0.5096648	0.64842849
2^{eme} sim	0.0	0.0	0.039952204	0.05082977
3^{eme} sim	0.63753396	0.8111119	0.664166	0.8449941

Tableau II.6 : résultat des 3 simulations de comparaison

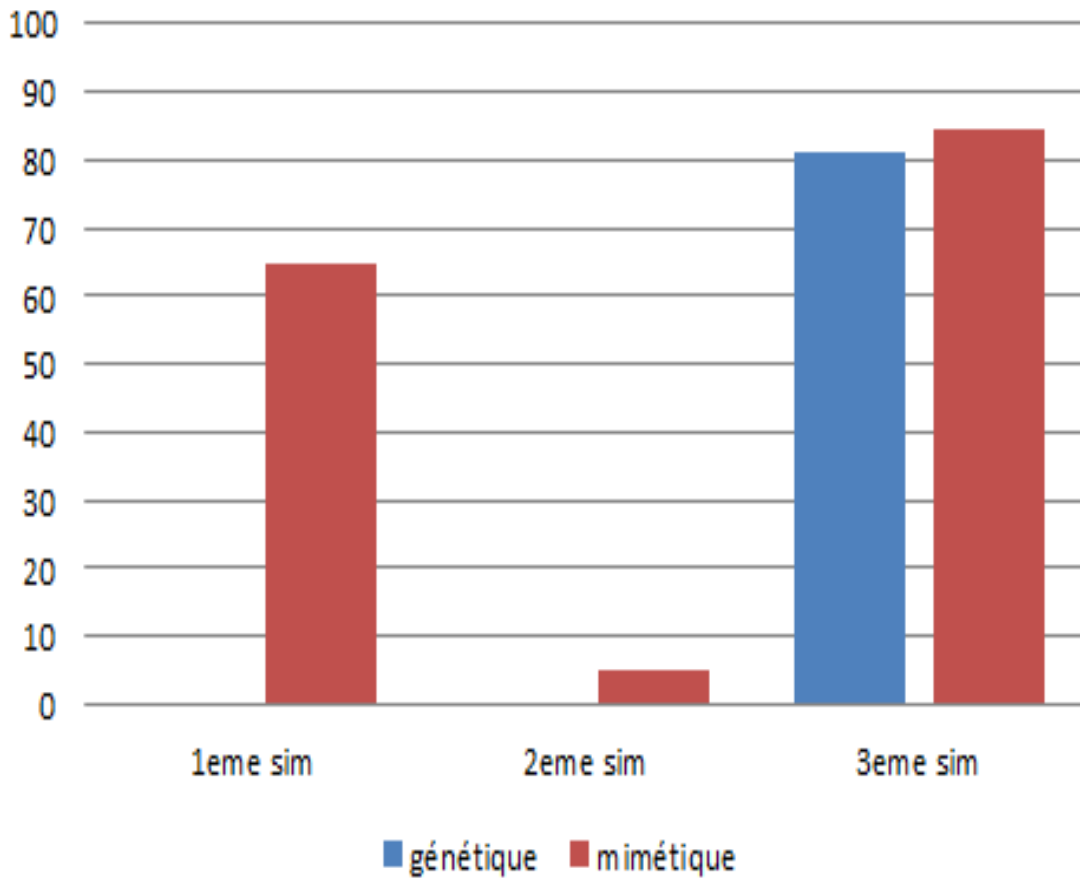


Figure II.7 : Histogramme de comparaison d'optimalité

V.4.1 Discussion

Malgré que l'algorithme génétique peut atteindre des pourcentages acceptables (3eme sim nous avons 81 %), nous remarquons que l'algorithme mimétique peut donner des taux plus élevés (64% première simulation et 84% troisième simulation).

Mais nous avons une petite charge de temps par rapport aux algorithmes génétique, mais elle est toujours linéaire en la comparant avec le temps d'exécution de la version génétique.

VI. Conclusion

Dans ce chapitre on a proposé un algorithme à base d'optimisation mimétique pour le problème de sélection de service web. Nous avons présenté les résultats de la validation de notre approche, afin d'évaluer l'efficacité de cette dernière.

Les résultats obtenus sont très acceptables et montrent leur supériorité par rapport à d'autres approches.

Conclusion générale

Ce projet de fin d'étude nous a permis d'instruire et améliorer nos connaissances.

Nous avons présenté dans ce mémoire les technologies liées aux services Web. Nous avons proposé aussi un algorithme de sélection qui se base sur l'algorithme mimétique pour l'optimisation des compositions.

Notre prototype sélectionne les compositions de services les plus satisfaisantes, en se basant sur cinq(05) critères de QOS : Latence, coût, fiabilité, disponibilité, 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

Comme perspectives à ce travail, nous proposons d'implémenter d'autres algorithmes de sélection, pour cela, nous pouvons citer :

- La programmation par contrainte.
- La programmation dynamique.
- les algorithmes d'optimisation à base de pareto
- ...

Références & Bibliographie

- [1] E. Cerami, Web Services Essentials, édition O'Reilly, Février 2002.
- [2] K. Hubert , M. Valérie , LES WEB SERVICES, Edition DUNOD, 2003.
- [3] Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler. 6 October 2000 (See <http://www.w3.org/TR/2000/REC-xml-20001006>.)
- [4] J-M. Chauvet, Services Web avec SOAP, WSDL, UDDI, ebXML..., Edition EYROLLES
- [5] SOAP Version 1.2 Part 2: Adjuncts, W3C Recommendation, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. Nielsen, 24 June 2003 (See <http://www.w3.org/TR/2003/REC-soap12-part2-20030624/>.)
- [6] N. Mitra et Y. Lafon, SOAP Version 1.2 Part 0: Primer (Second Edition 2003).
- [7] M. Dali Yahia, Selection des services web a base de QoWS, Mémoire de master spécialité MID , UABT,2011
- [8] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft, R. Chinnici, M. Gudgin, J-J. Moreau, J. Schlimmer, S. Weerawarana, 10 November 2003 (See <http://www.w3.org/TR/2003/WD-wsdl20-20031110/>.)
- [9] <http://www.w3schools.com/wsdl/default.asp>
- [10] Tableau tire et traduit par [M. Leblanc, HEC Montréal, 14 novembre 2002] de Eric Newcomer, Understanding Web Services : XML, WSDL, SOAP and UDDI. Éd.Addison-Wesley, 2002 p.25
- [11] Microsoft, Ariba, et IBM premier version (mai 2001) (http://fr.wikipedia.org/wiki/Universal_Description_Discovery_and_Integration#Historique)
- [12] <http://www.benoitpiette.com/labo/introduction-aux-web-services/uddi.php3.htm>
- [13] J-M. Chauvet, Services Web avec SOAP, WSDL, UDDI, ebXML, Eyrolles, 2002.
- [14] A. Vezain, Les service web - présentation générale, rapport technique, Association HERMES, Février 2005.
- [15] M. Vialette, Web Services Communication inter langage, Version 2.0 Ecole Supérieur d'Informatique de Paris, 8 mars 2006.
- [16] Wikipedia®, http://fr.wikipedia.org/wiki/Service_web

- [17] <http://www.techno-science.net/?onglet=glossaire&definition=1462>
- [18] H. Djelmoudi, M. Belaredj, la Sélection de services web à base des systèmes immunitaires artificiel (CLONALG), Mémoire de master spécialité MID, 2012
- [19] E. Alrifai, T. Risse Combining Global Optimization with local selection for Efficient QoS-aware Service Composition In WWW09, April 20-24, 2009, Madrid, Spain.
- [20] E. Alrifai, T. Risse Selecting Skyline Services for QoS-based Web Service Composition In Proceedings of the WWW 2010, April 26-30, 2010, Raleigh, North Carolina, USA.
- [21] Q. Yu, A. Bouguettaya, Foundations for Efficient Web Service Selection Springer Science+Business Media, 2010.
- [22] M. M. Akbar, E. G. Manning, G. C. Shoja, and S. Khan. Heuristic solutions for the multiple-choice multi-dimension knapsack problem. In Proceedings of the International Conference on Computational Science-Part II, pages 659-668, London, UK, 2001. Springer-Verlag.
- [23] D. Ardagna and B. Pernici. Global and local QoS constraints guarantee in web service selection. In Proceedings of the IEEE International Conference on Web Services. Pages 805-806, Washington, DC, USA, 2005. IEEE Computer Society.
- [24] D. Ardagna and B. Pernici Adaptive service composition in flexible processes. IEEE Transactions on Software Engineering, 33(6): 369-384, 2007 Dustdar, S and Schreiner, W. “ A survey on web services composition”, Int. J. Web and Grid Services, Vol. 1, No. 1, pp.1-30. (2005).
- [25] F. Li, F. Yang, K. Shuang and S. Su. Q-peer: A decentralized QoS registry architecture for web services. In Proceedings of the International Conference on Services Computing, pages 145-156, 2007.
- [26] Z. Belmebrouk, M. Benyettou Les Algorithmes Mimétique, 2011-2012, pages 3-5
- [27] J-K. Hao, Recherche local et mimétique : de la théorie à la pratique, Université d'Angers

[28] Z. Belmebrouk, M. Benyettou Les Algorithmes Mimétique, 2011-2012, pages 8

[29] F. LARDEUX, Approches hybrides pour les problèmes de satisfiabilité (SAT et MAX-SAT), Thèse de doctorat, Ecole Doctorale d'Angers

