

N° d'ordre :
N° de série :

République Algérienne Démocratique et Populaire



**Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique**

UNIVERSITÉ ECHAÏD HAMMA LAKHDAR D'EL OUED

FACULTÉ DES SCIENCES EXACTES

DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'étude

MASTER ACADEMIQUE

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Systèmes Distribués et Intelligence Artificiel (SDIA)

Thème

**Sélection de Service Web à base
d'algorithme Mimétique**

Présenté par: HIMA Faiçal & GAID Hamza

Soutenu : **Juin 2016**

Année universitaire 2015 - 2016

Remerciements

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

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

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

Dédicace

A la mémoire de mon père

A mon mère

A ma femme

A Mes enfants

A mes frères et sœurs

A tous mes ami(e)s

HIMA Faiçal

Dédicace

A mes parents

A mes frères et sœurs

A mes grands parents

A tous mes ami(e)s

GAID Hamza

Résumé

Le nombre de services Web croit à une vitesse explosive, créant ainsi, de grands défis dans l'extraction exacte, efficace et automatique de services cibles pour les utilisateurs.

Ce projet propose un algorithme mémétique qui combine un algorithme génétique et une méthode de recherche locale pour résoudre ce problème.

Mots clés

Services Web, Recherche et Sélection de Services, algorithme mimétique.

Abstract

The number of Web services believed to explosive speed, this creating great challenges in extracting accurate, efficient and automatic target services to users.

This project proposes a memetic algorithm that combines genetic algorithm and a local search method to solve this problem.

Keywords

Web services, Search and selection services, memetic algorithm

ملخص :

عدد من خدمات الويب يحتاج سرعة قوية، وبالتالي خلق تحديات كبيرة في استخراج الخدمات المستهدفة للمستخدمين حتى تكون دقيقة وفعالة وآلية .
ويقترح هذا المشروع خوارزمية ميميتيك التي تجمع بين الخوارزمية الجينية وطريقة بحث المحليين على حل هذه المشكلة.

كلمات مفتاحية :

خدمات الشبكة العالمية، اختيار البحوث والخدمات، خوارزمية ميميتيك.

Table des matières

Résumé.....	I
Table des matières.....	II - III
Liste des figures	IV
Liste des tableaux	V
Introduction générale	01
I. Contexte.....	02
II. Contribution.....	02
III. Plan du mémoire.....	03
Partie 01 Etat de l'art	04
Chapitre I a technologie des Services Web	05
I.1. Introduction	06
I.2. Définitions	06
I.3.Pourquoi les services Web	07
I.4.Fonctionnement	07
I.5. Architecture	08
I.5.1. La couche de publication	09
I.5.2. La couche description	09
I.5.3. La couche communication	09
I.5.4. La couche transport	09
I.6. Les Standards	10
I.6.1.XML, la langue maternelle des services web	10
I.6.2. Le protocole SOAP (simple Object Access Protocol)	10
I.6.2.1.Présentation	10
I.1.6.2.2. Fonctionnement	11
I.6.3. WSDL	12
I.6.3.1.Structure et description WSDL	12
I.6.4.UDDI	13
I.6.4.1.Principe	13
I.6.4.2.Consultation de l'annuaire	13
I.7.Conclusion	14
Chapitre02 La Sélection des services web	15
II.1. Introduction	16
II.2. Définition	16
II.3. Propriétés des services web	16
II.3.1. Les propriétés fonctionnelles	16
II.3.2. Les propriétés non fonctionnelles.....	16
II.4. Approches de sélection.....	17
II.4.1.Sélection basée sur les besoins fonctionnels.....	17
II.4.2 Sélection basée sur les besoins non fonctionnels.....	18
II.4.3 Sélection basée sur la qualité de service.....	19
II.4.3.1. La sélection mono-objective.....	20
A. Sélection Locale.....	21
B. Sélection globale.....	21
B.1.L'optimisation globale exacte (la programmation par contrainte ou le	

Énumérations exhaustives)	22
B.2.L'optimisation globale approximative (<i>méta-heuristiques</i>)	22
C. La sélection hybride	23
II.4.3.2. La sélection multi-objective.....	24
II.4.3.3. La sélection mono et multi-objective.....	24
II.5.Problématique.....	24
II.6.Conclusion.....	24
Partie 02 Contribution	25
Chapitre 03 Conception.....	26
III.1.Introduction.....	27
III.2.Description de la base.....	27
III.2.1 Description de la base.....	27
III.2.2 Description de la requête.....	27
III.3.Algorithme Mimétique.....	28
III.3.1 Définition.....	28
III.3.2 Algorithme génétique.....	28
III.3.2.1 Codage d'individus.....	28
III.3.2.2 Fonctions d'évaluation.....	29
A. Critères de QoS considérés.....	29
B. Fonctions d'agrégation.....	29
C. Fonction objective.....	30
III.2.3.3 Opérateurs génétiques.....	30
III.3.3 Algorithme de recherche locale.....	31
III.3.3.1.L'algorithme mémétique.....	32
III.3.3.2.Description de l'algorithme mimétique	32
III.4.Conclusion.....	34
Chapitre IV Implémentation et Expérimentations.....	34
IV.1. Introduction.....	35
IV.2. Présentation des outils technologiques utilisés.....	35
IV.3.Implémentation.....	36
IV.3. Présentation de l'Interface Humain/Machine (IHM).....	36 - 38
IV.4. Expérimentation	38
IV.4.1. Première configuration	38
IV.4.2. Deuxième configuration	38
IV.4.3. Troisième configuration	39
IV.4.4. Quatrième configuration	39
IV.4.5. Discussion.....	39
IV.4.6. Comparaison génétique / mimétique.....	39
IV.5.Conclusion.....	40
Conclusion Générale.....	41
Références Bibliographiques.....	42

Liste des figures

Figure I.1 : Fonctionnement des services Web [BKK,04].....	07
Figure I.2 : Architecture en pile des services web [BKK, 04]	08
Figure I.3 : Traitement d'un message SOAP [BO,12]	11
Figure I.4 : Structure d'un document WSDL [TLH,10]	12
Figure I.5 : Schéma général de l'annuaire UDDI [BO,12]	14
Figure II.1 : les approches de sélection de services web.....	20
Figure III.1 : Codage d'un service composite.....	28
Figure III.2 : Sélection globale.....	31
Figure III.3 : l'algorithme mimétique.....	32
Figure IV.1 : Fenêtre principale du système de sélection de SW.....	36
Figure IV.2 : Base des services Web	36
Figure IV.3 : Les tâches d'utilisateur	37
Figure IV.4 : Les paramètres d'algorithme génétique	37
Figure IV.5 : Les critères de QoS	37
Figure IV.6 : Les opérations générales.....	38

Liste des tableaux

Tableau II.1 : Bilan des approches de sélection basées sur l'aspect fonctionnel.....	18
Tableau II.2 : Bilan des approches de sélection basées sur l'aspect fonctionnel.	19
Tableau II.3 : Bilan des approches de La sélection locale.....	21.
Tableau II.4 : Bilan des approches de l'optimisation globale exacte.	22
Tableau II.5 : Bilan des approches de l'optimisation globale approximative (méta-heuristiques).	23
Tableau II.6 : Bilan des approches de La sélection hybride.	23
Tableau III.1 : Les fonctions d'agrégation des critères de QoS.	29
Tableau III.2 : Les intervalles des critères de QoS [BD,13].	30
Tableau IV.1 : Résultats de la 2 ^{ième} configuration	38
Tableau IV.2 : Résultats de la 3 ^{ième} configuration.	39
Tableau IV.3 : Résultats de la 4 ^{ième} configuration.	39
Tableau IV.4 : résultat de comparaison (algorithme génétique / mimétique)	40

Introduction Générale

I. Contexte

Le Web a évolué pour englober diverses sources d'information accessible mondialement. Les organisations de tous les spectres ont déjà déplacé leurs opérations principales au Web, ce qui a entraîné une croissance rapide des différentes applications Web et en particulier les services web. Ces derniers ont l'avantage de faciliter l'intégration d'applications ou de systèmes d'information.

L'importance des standards dans ce contexte a sans doute accentué le phénomène. Le service web traite les données et relie à l'internet et à l'externe d'une application logiciel. Il a maintenant évolué pour englober diverses ressources d'information dans le monde entier accessibles.

II. Problématique

Le problème de la sélection des services web a attiré l'attention des chercheurs au cours de la dernière décennie. La raison à 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.

III. Contribution

On propose dans ce travail, une approche de sélection de services Web composites à base d'algorithme Mimétique.

IV. Plan du mémoire

Ce mémoire est découpé en deux parties :

- ✓ **La 1^{ière} Partie « l' état de l'art » :**
 - **Chapitre I (la technologie des services Web) :** qui donne un aperçu sur les concepts de base du paradigme de services web.
 - **Chapitre II (la sélection de services web) :** qui donne un survol sur les approches proposées dans la littérature.

- ✓ **La 2^{ième} Partie «Contribution » :**
 - **Chapitre III (Conception) :** ce chapitre montre la conception de notre système.
 - **Chapitre IV (Implémentation et Expérimentations) :** nous décrivons un prototype implémentant notre application ainsi que les résultats obtenus.
 - **La conclusion générale :** résume les résultats de notre travail, et présente les perspectives que nous souhaitons réaliser dans le futur.

Partie 01

Etat de l'art

Chapitre I :La technologie des Services Web

I.1. Introduction

Les Web Services prennent leur origine dans l'informatique distribuée et dans l'avènement du Web. Ils poursuivent un vieux rêve de l'informatique distribuée où les applications pourraient inter opérer à travers le réseau, indépendamment de leur plateforme et de leur langage d'implémentation.

Les Web services regroupent tout un ensemble de technologies bâties sur des standards. Ils permettent de créer des composants logiciels distribués, d'écrire leur interface et de les utiliser indépendamment de la plateforme sur laquelle ils sont implémenté

I.2. Définitions

Dans la littérature, il existe de nombreuses définitions des Web services. Cette prolixité montre que la notion de service Web a besoin d'être éclaircie, et motivée des travaux de recherches.

En 2001, Nagy et al. Écrivent dans [CNW01] : "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."

En 2002, O'Sullivan Justin, Edmond David, Ter Hofstede Arthur écrivant dans [JET02] : "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."

Le consortium W3C définit un service Web comme étant : "une application ou un composant logiciel identifié par un URI, dont ses interfaces et ses liens peuvent être décrits en XML, sa définition peut être découverte par d'autres services Web et il peut interagir directement avec d'autres services Web à travers le langage XML et en utilisant des protocoles Internet standards ". [CGS,03]

I.3. Pourquoi les services Web

1. Les services Web permettent d’interconnecter :
 - ✚ Différentes entreprises
 - ✚ Différents matériels
 - ✚ Différentes applications
 - ✚ Différents clients
2. réutilisation dans un environnement ouvert (runtime)
3. Distribuer et intégrer des logiques métiers Vers le Web sémantique
4. Les services Web sont faiblement couplés

I.4. Fonctionnement

Le fonctionnement des services Web s'articule autour de trois acteurs principaux illustrés illustré sur la **Figure I.1**.

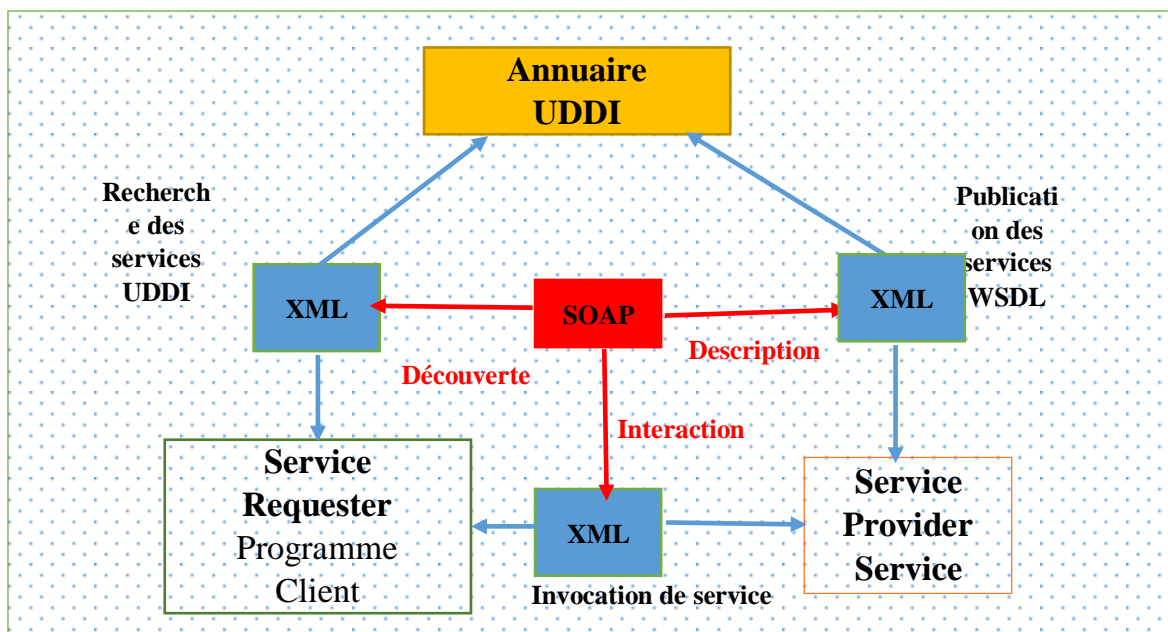


Figure I.1 : Fonctionnement des services Web [BKK,12]

Service provider service: Le fournisseur de service met en application le service Web et le rend disponible sur Internet.

Service requester programme client: C'est n'importe quel consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (REST, XML-RPC, SOAP).

Annuaire service registry: Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver. Les interactions entre ces trois acteurs suivent plusieurs étapes :

1. La publication du service : le fournisseur diffuse les descriptions de ses services Web dans l'annuaire.
2. La recherche du service : le client cherche un service particulier, il s'adresse à un annuaire qui va lui fournir les descriptions et les URL des services demandés afin de lui permettre de les invoquer.
3. L'invocation du service : une fois que le client récupère l'URL et la description du service, il les utilise pour l'invoquer auprès du fournisseur de services.

I.5. Architecture

L'architecture des services web est implémenté à l'aide des diverses technologies. Elle est organisée en quatre couches comme illustré sur la **Figure I.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 etc. Comme représenter sur la **Figure I.2** dans un souci d'interopérabilité les déférentes couche de la pile des

Services web s'interfacent avec des standards. Les fonctions de couches supérieures reposent sur celles de couches inferieurs.

La technologie des services Web offre de fortes potentialités pour surmonter les problèmes d'interopérabilité des systèmes. Elle constitue un cadre prometteur pour l'intégration des applications, et pour la gestion des interactions entre divers partenaires dans un environnement distribués, hétérogènes, ouvert et versatile qui est le Web

[BKK04].

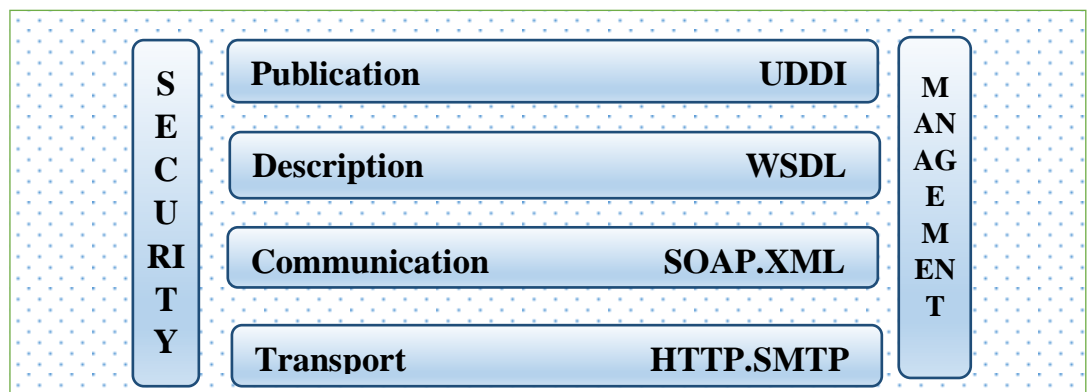


Figure I.2 : Architecture en pile des services web [BKK04]

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

I.5.1. La couche de publication

Repose sur le protocole UDDI (Universal Description, Discovery and Integration). Est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients.

I.5.2. La couche description

Est prise en charge par le langage WSDL (Web Service Description Language) [BK,12], est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

I.5.3. 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 un protocole standard de communication. C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes. Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

I.5.4. 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

I.6. Les Standards

I.6.1.XML, la langue maternelle des services web

XML est l'acronyme de "Extensible Markup Language", expression signifiant "langage de balisage extensible".

XML est un standard promulgué par le W3C, l'organisme chargé de standardiser les évolutions du Web. On retrouve dans XML une généralisation des idées contenues dans HTML, on n'utilise les balises que pour décrire l'aspect graphique que doit revêtir la page dans le navigateur Web. Dans XML, les balises permettent d'associer toute sorte d'informations au fil du texte. [BK,12].

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 [BK,12].

Le contenu d'un document XML est décrit par une succession d'«éléments», blocs de texte encadré par des paires de balises ouvrantes et fermantes, qui sont les « unités de contenu ». Ces éléments sont liés entre eux par une hiérarchie, certains éléments apparaissant imbriqués dans d'autres. XML et DTD permettent aussi la séparation effective du contenu de la structure de document.

I.6.2. Le protocole SOAP (simple Object Access Protocol)

I.6.2.1.Présentation

Beaucoup de définitions normalisées de SOAP ont été proposées. Une particulièrement intéressante définit SOAP comme étant une spécification pour une omniprésence, basée sur XML et sur des infrastructures distribuées. [BO,12].

Spécification Car SOAP est un document qui définit le modèle de communication. L'idée de base est que si les deux parties ont créé des programmes de mêmes spécifications, ils seront en mesure d'interagir de façon transparente.

- **Omniprésente** car SOAP est défini à un niveau suffisamment élevé d'abstractions que tout système d'exploitation et combinaison de langages de programmation peuvent être utilisés pour créer des programmes compatibles SOAP.
- **Basé sur XML**, SOAP est construit sur XML, ce qui signifie que les documents SOAP sont des documents XML construits en fonction d'un cahier de charges plus strict.
- **Infrastructure distribuée**, SOAP ne précise pas quelles données peuvent être déplacées ou bien quels appels de fonctions peuvent avoir lieu sur elle. Les applications construites sur la spécification SOAP peuvent déplacer les données d'un ordinateur A à un ordinateur B et par la suite à une autre application écrite sur la même spécification.

I.1.6.2.2. Fonctionnement

La requête SOAP transmise par le client à l'aide des protocole de transmissions (exemple HTTP) passe par un Listener() que le permet de adresser (cas de java) aux bibliothèques, ces bibliothèque donnée au client l'accède aux méthode des services. (Figure I.3) .

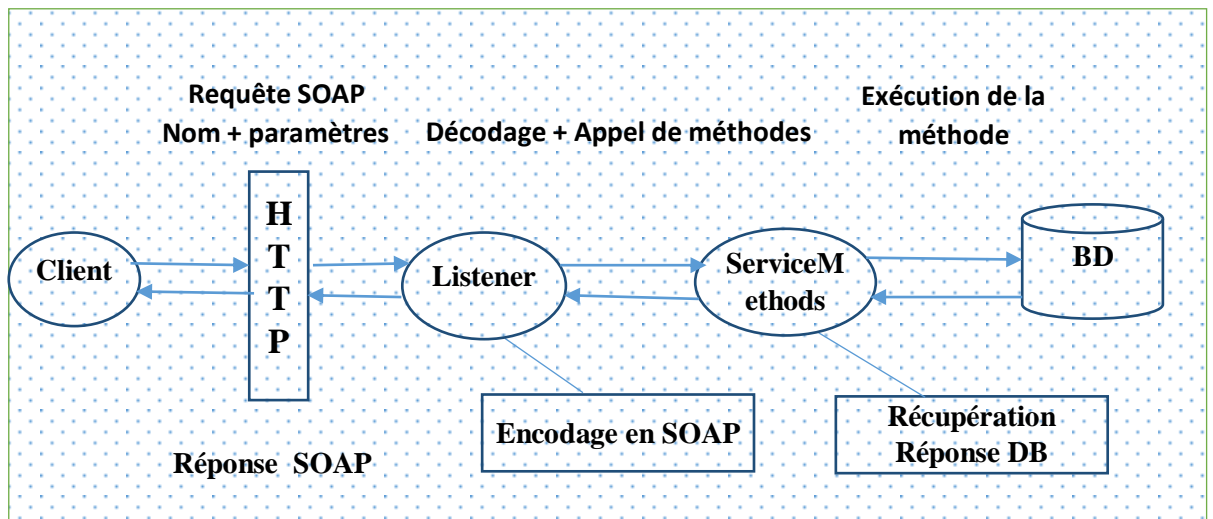


Figure I.3 : Traitement d'un message SOAP [BO,12]

Si l'exécution des méthodes a besoin a des informations de la base de donnée alors il doit connecter avec la base de donnée [BO,12] .

I.6.3. WSDL

WSDL [TLH,10] a été développé conjointement par IBM, Microsoft et Ariba 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.

I.6.3.1. Structure et description WSDL

Un document WSDL se compose d'un ensemble d'éléments décrivant les types de données utilisés par le service, les messages que le service peut recevoir, ainsi que les liaisons SOAP associées à chaque message. Le schéma suivant illustre la structure du langage WSDL qui est un document XML, en décrivant les relations entre les sections constituant un document WSDL. **Figure I.4** [TLH,10]

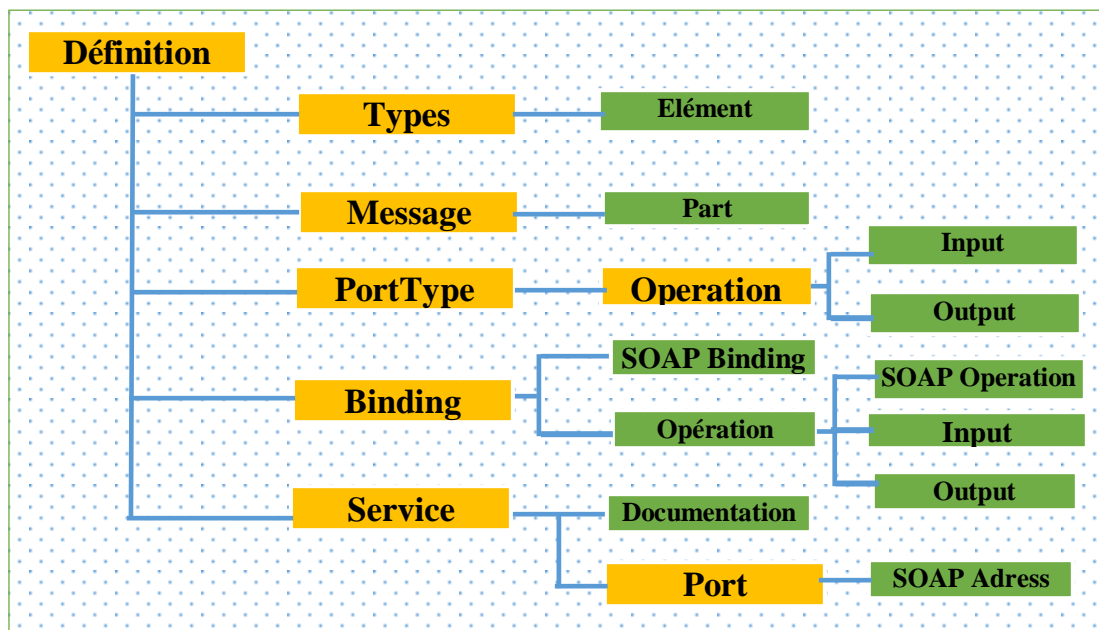


Figure I.4 : Structure d'un document WSDL [TLH,10]

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.

I.6.4.UDDI

I.6.4.1.Principe

L'annuaire des services UDDI [BO,12] est un standard pour la publication et la découverte des informations sur les services Web. La spécification UDDI est une initiative lancée par ARIBA, Microsoft et IBM. Cette spécification n'est pas gérée par le W3C mais par le groupe OASIS. La spécification UDDI vise à créer une plate-forme indépendante, un espace de travail (frame work) ouvert pour la description, la découverte et l'intégration des services des entreprises.

I.6.4.2.Consultation de l'annuaire

L'annuaire UDDI se concentre sur le processus de découverte de l'architecture orientée services (SOA), et utilise des technologies standards telles que XML, SOAP et WSDL qui permettent de simplifier la collaboration entre partenaires dans le cadre des échanges commerciaux. L'accès au référentiel s'effectue de différentes manières.

- Les pages blanches comprennent la liste des entreprises ainsi que des informations associées à ces dernières (coordonnées, description de l'entreprise, identifiants...).
- Les pages jaunes recensent les services Web de chacune des entreprises sous le standard WSDL.
- Les pages vertes fournissent des informations techniques précises sur les services fournis.

Les entreprises publient les descriptions de leurs services Web en UDDI, sous la forme de fichiers WSDL. Ainsi, les clients peuvent plus facilement rechercher les services Web dont ils ont besoin en interrogeant le registre UDDI.

Lorsqu'un client trouve une description de service Web qui lui convient, il télécharge son fichier WSDL depuis le registre UDDI. Ensuite, à partir des informations inscrites dans le fichier WSDL, notamment la référence vers le service

Web, le client peut invoquer le service Web et lui demande d'exécuter certaines de ses fonctionnalités.

Le scénario classique d'utilisation de UDDI est illustré ci-dessous. L'entreprise B a publié le service Web S, et l'entreprise A est client de ce service **Figure I.5 [BO,12]**

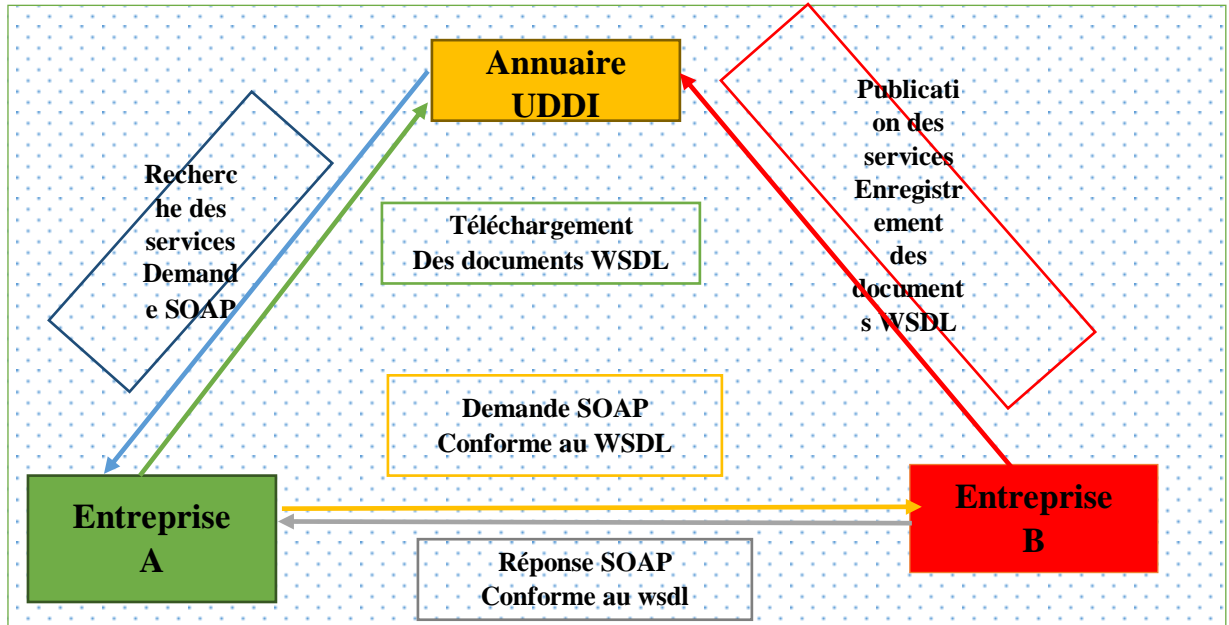


Figure I.5 : Schéma général de l'annuaire UDDI [BO,12]

I.7.Conclusion

Dans ce chapitre, on a présenté une formalisation du problème de sélection automatique de services web, ensuite on a montré une partie d'état de l'art des différentes approches de sélection de services Web composites proposées dans la littérature pour résoudre ce problème. Le chapitre suivant est consacré à étudier Sélection des services web

Chapitre II

La sélection des Services Web

II.1. Introduction

Avec l'augmentation croissante du nombre de fournisseurs de services web, il arrive très fréquemment que de nombreux services web répondent à un même besoin fonctionnel, mais qui offrent des propriétés de qualité de services différentes, une sélection doit être alors réalisée pour déterminer quels sont les services web pertinents qui satisferaient les besoins d'un utilisateur.

Dans la suite de ce chapitre, nous passerons en revue les principaux travaux proposés dans ce domaine.

II.2. Définition

La sélection des services web consiste à choisir parmi un ensemble de services web découverts ceux le service le plus adéquat qui répondent aux préférences de l'utilisateur sur la base des besoins fonctionnels et non fonctionnels.

II.3. Propriétés des services web

Les propriétés fonctionnelles et non fonctionnelles de services définissent un service web. En effet, les services se distinguent par les fonctionnalités qu'ils peuvent fournir. Ainsi deux services similaires, c'est à dire qu'ils offrent tous les deux les mêmes fonctionnalités, peuvent se différencier par leurs propriétés non fonctionnelles.

II.3.1. Les propriétés fonctionnelles

Les propriétés fonctionnelles d'un service désignent les opérations qu'il peut fournir. Les propriétés fonctionnelles sont décrites dans la description de service en termes d'entrées, sorties, pré-conditions et effets du service qui reflètent le fonctionnement du service.

II.3.2. Les propriétés non fonctionnelles

Les propriétés non fonctionnelles de service appelées aussi : qualités de service définissent les capacités de service à fonctionner dans de bonnes conditions en termes de disponibilité, performance, coût d'invocation, fiabilité, etc.

✓ Qualité de Service

La qualité de service(QoS) représente l'aptitude d'un service à répondre d'une manière adéquate à des exigences, exprimées ou implicites, qui visent à satisfaire ses usagers. Ces exigences peuvent être liées à plusieurs aspects d'un service, par exemple:

- **Le débit** : Le nombre de requêtes servies pendant un intervalle de temps.
- **Le temps de réponse** : Le temps requis pour compléter une requête du service web.
- **La fiabilité (sûreté)** : La capacité d'un service d'exécuter correctement ses fonctions.
- **La scalabilité** : La capacité du service de traiter le plus grand nombre d'opérations ou de transactions pendant une période donnée tout en gardant les mêmes performances.
- **La robustesse** : La probabilité qu'un service peut réagir proprement à des messages d'entrée invalides, incomplètes ou conflictuelles.
- **La disponibilité** : La probabilité d'accessibilité d'un service.
- **Le prix d'exécution** : C'est le prix qu'un client du service doit payer pour bénéficier du service.
- **La réputation** : C'est une mesure de la crédibilité du service.
- **La sécurité** : C'est un regroupement d'un ensemble de qualités à savoir: la confidentialité, le cryptage des messages et le contrôle d'accès.

II.4. Les approches de sélection

II.4.1. Sélection basée sur les besoins fonctionnels

Dans ce cas, la sélection de service se base principalement sur les propriétés fonctionnelles d'un service. Un fournisseur de service publie une description de son service avec laquelle un utilisateur peut chercher le service qui répond aux fonctionnalités requises. Les moyens de sélection de services actuels reposent sur l'adéquation syntaxique exacte entre l'interface de service offerte par les fournisseurs et l'interface de service requise par l'utilisateur.

Cette technique de recherche syntaxique s'est avérée faible. Les fonctionnalités offertes sont décrites de manière hétérogène et le résultat peut ne pas correspondre aux attentes des clients, car c'est une recherche basée sur les mots clés. De nombreux travaux prennent par conséquent l'hypothèse que la sélection s'effectue sur une base des besoins fonctionnels. **(Voir Tableau II.1)**

	Etudes	Outils	Caractéristiques
Approches basés sur la description sémantique	[W3C,05]	- La spécification SAWSDL. - Annotation sémantique. - Ontologies.	- expressivité efficacité du langage sémantique. - Impacte de la profondeur des graphes des ontologies sur les temps de raisonnement.
	[MP,08]	- SAWSDL-MX. - Matchmaker.	-expressivité efficacité du langage sémantique. - Impacte de la profondeur des graphes des ontologies sur les temps de raisonnement.
Approches basés sur la méthode d'analyse de concepts formels	[ZMP,08]	- WSPAB. - Filtration progressive (Nombre de paramètre d'entrées comme facteur de sélection).	- Détermine des substituts de service. - Similarité des opérations basée sur le nombre de paramètres d'entrées.
	[LMG,06]	- WordNet pour faire une analyse morphologique.	- Utilisation des mots-clés.
	[DSXA,05]	- TF/IDF pour la recherche d'informations	-Relève le problème de taille de treillis en fusionnant les opérations similaires.

Tableau II.1 : Bilan des approches de sélection basées sur l'aspect fonctionnel[GH,12]

II.4.2 Sélection basée sur les besoins non fonctionnels

Pour réaliser la sélection parmi un ensemble de services dont les fonctionnalités sont équivalentes, seules les propriétés non fonctionnelles du service font l'objet de raffinement pour sélectionner le meilleur service. Les besoins non fonctionnels des Web services sont généralement exprimés à l'aide des critères de qualité de service.

Dans cette sélection, il y a plusieurs travaux de recherche s'y intéressent. (Voir Tableau II.2)

	Etudes	Outils	Caractéristiques
Approches basés sur la description sémantique	[XTMI,06]	- WSMO-QoS	- ontologie complémentaire fournit plus de détaille sur les aspects non-fonctionnels - modèle de propriétés non-fonctionnelles extensible
	[EQ,07]	- Gestionnaire de propriétés non-fonctionnelles - WS-QoSMan	- Séparation des descriptions de services et des informations reliées aux propriétés non-fonctionnelles. - Modèle de propriétés non-fonctionnelles n'est pas extensible.
Approches basés sur des mécanismes de réputation et du vote	[YJ,07] (une étude des approches)	- Gestionnaire de votes	- Observation et historique d'un service. - Coopération des utilisateurs.
Approches basés sur une modélisation graphique et analytique	[NSEN,05]	- Arbre de recherche - Algorithme K-mean	- Heuristique - un modèle de propriétés non-fonctionnelles générique. - l'évaluation d'utilité de service par un algorithme de clustering.
	[MJ,07]	- Arbregénétique	- Heuristique - Modèle d'agrégation de propriétés non-fonctionnelles .
	[YHX,06]	- Algorithme de routage pour la représentation - l'algorithme de dijkstra pour la sélection	- la sélection d'un service individuel est influencée par la sélection de tous les services de la composition de services .

Tableau II.2 : Bilan des approches de sélection basées sur l'aspect non-fonctionnel

II.4.3 Sélection basée sur la qualité de service

✓ Modèles de QoS existants

Il n'y a pas un consensus bien précis au sujet de l'ensemble des QoS importantes pour les services web, le groupe de travail « Architecture des Services Web » du W3C travaillant sur les architectures des services web, a identifié et décrit un ensemble de paramètres de QoS pour les services web à savoir : la performance (qui englobe le débit, le temps de réponse et le temps d'exécution), la fiabilité, la scalabilité ou l'adaptation au facteur d'échelle, la capacité, la robustesse, le traitement d'exception, l'exactitude, l'intégrité, l'accessibilité, la disponibilité, l'interopérabilité et la sécurité.

La plupart des travaux de recherche qui ont essayé d'identifier et de classer les paramètres de QoS ont pris en considération les paramètres définis par le W3C aux quels sont associés dans certains travaux d'autres paramètres.

✓ **La fonction objective**

Le rôle de cette fonction est maximisée les critères positifs (Réputation, Disponibilité et Fiabilité.....) et minimisée les critères négatifs (coût et temps d'exécution.....) .

✓ La sélection des services web basée sur la qualité de service a fait l'objet de plusieurs travaux, de façon générale on distingue 03 grandes classes. (Voir Figure II.1)

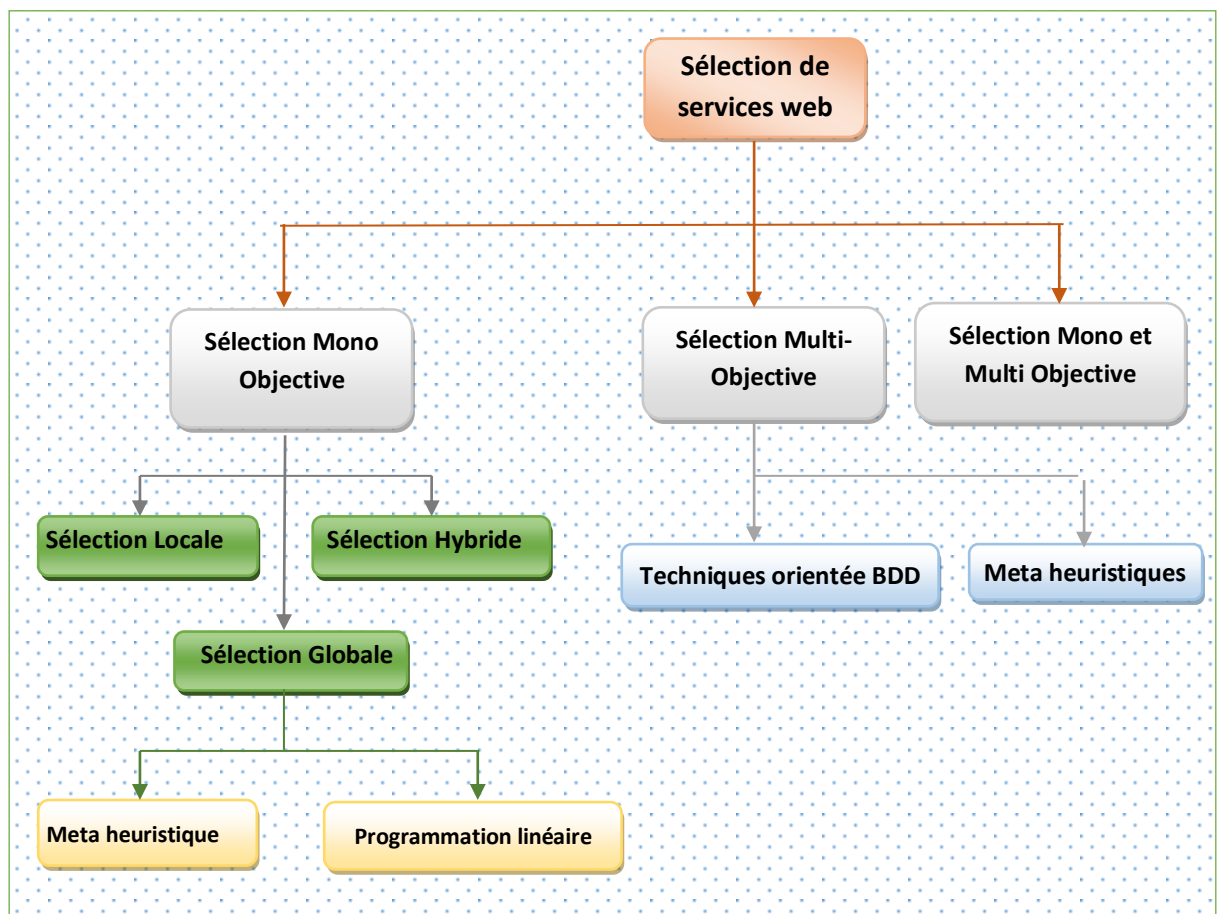


Figure II.1:les approches de sélection de services web

II.4.3.1. La sélection mono-objective

On considère une seule fonction objective qui associe des poids aux différents attributs de QoS (tout les valeurs de qualité de service soient agrégés en un seul score), elle compose en 03 sous approches (classes) .

A. Sélection Locale

Elle a pour objectif de choisir le meilleur Web service pour chaque tâche individuelle à part entière en considérant des contraintes de QoS relatives à chaque tâche plutôt qu'en considérant des contraintes de QoS globales exprimées pour l'ensembles des tâches. Il s'agit de sélectionner pour chaque tâche un service web apte à l'exécuter en tenant en compte les contraintes de l'utilisateur.

Les études de cette sélection sont résumées dans le **Tableau II.3**

	Etudes	Outils	Caractéristiques
La selection locale	[BS, 02]	- les contraintes globales	<ul style="list-style-type: none"> - adapter aux environnements distribuée . - la gestion de laQoS (mesures, mise à jours ...) fait par des facilitateurs (brokers) distribués - possède une complexité linéaire .
	[MT, 09]	- les contraintes globales	<ul style="list-style-type: none"> - diviser les contraintes globales en contraintes locales basant sur la distribution statistiques des valeurs de QoS et gèrent la séquence
	[KDAM, 11 a]	- les contraintes globales	<ul style="list-style-type: none"> - sélection de services web qui satisfaisant les préférences des sorties de services. - appliquent un tri local (pour chaque classe) - adoptant la notion de fuzzy dominance pour la comparaison entre les services . - retiennent les K premières solutions Les auteurs ne gèrent pas les contraintes globales.

Tableau II.3: Bilan des approches de la sélection locale[TLH,10]

B. Sélection globale

Contrairement à la méthode précédente, on choisit la combinaison de services web qui garantit la meilleure qualité globale en tenant compte des contraintes de QoS et des préférences globales assignées pour l'ensemble des tâches.

A fin de pouvoir appliquer une stratégie de sélection globale, on calcule la valeur des critères de QoS relative au service composite. Le calcul de ces critères s'appuie sur l'application de fonctions d'agrégation. Ensuite, on choisit la combinaison qui offre les meilleures valeurs de QoS parmi toutes les combinaisons possibles par rapport aux contraintes et aux préférences de l'utilisateur.

B.1.L'optimisation globale exacte (la programmation par contrainte ou les énumérations exhaustives)

Ces méthodes sont basées sur la programmation entière en temps d'exécution exponentiel pour donner des résultats optimaux, il existe plusieurs travaux faisant dans cette optimisation résumée en (Tableau II.4)

	Etudes	Outils	Caractéristiques
L'optimisation globale exacte	[KL, 09]	programmation entière	- donnent des résultats optimaux.
	[LBMJ,03]	programmation entière et mixtes MIP	- donnent des résultats optimaux
	[NW, 88]	programmation entière et mixtes MIP	-trouver la composition optimale des services
	[DG , 05]	programmation linéaire avec d'inclure les contraintes locales	- les contraintes globales sont exprimées sur la composition entière par l'utilisateur final. - les contraintes locales sont spécifiées par le concepteur de la composition au niveau des classes
	[ZL, 09]	un algorithme MIP avec remplacement les services en pannes	- la réparation doit garantir la satisfaction des contraintes globales de la composition
	[Ma, 03]	programmation entière	- programmation entière est efficace si le nombre de service est petit . - le temps d'exécution n'est plus raisonnable si le nombre de service dépasse une certaine limite .

Tableau II.4: Bilan des approches de l'optimisation globale exacte[TLH,10]

B.2.L'optimisation globale approximative (méta-heuristiques)

Elle adopte des recherches heuristiques ou des méta-heuristiques (algorithmes d'optimisation génériques, qui adoptent une recherche locale) pour donner des résultats proches de l'optimal, tout en ayant un temps d'exécution abordable (polynomial), on cite quelques études de l'optimisation globale approximative en (Tableau II.5)

	Etudes	Outils (heuristique utilisée)	Caractéristiques
L'optimisation globale approximative (méta-heuristiques)	[Kh, 88]	- UHE	- utilise la mesure (la consommation des ressources agrégés) - mettre à jour un élément de chaque groupe à chaque tour de sélection
	[ME, 01]	- M-UHE	- propose une étape de prétraitement de trouver une solution réalisable et une étape de post-traitement - améliorer la valeur totale de la solution
	[ME, 06]	- C-UHE	- évaluent la performance et l' optimalité de l'heuristique M-UHE - meilleure par rapport à M-UHE en termes de temps d'exécution et en termes de degré d'optimalité - fonctionne mieux dans le cas où l'objectif à maximiser
	[TYK, 07]	- WS-UHE	- complexité temporelle polynomiale - applique des améliorations sur une composition de services - les valeurs de QoS proviennent de plusieurs facilitateurs - chaque facilitateur gère une classe donnée
	[KE, 95]	-SPO	- optimisation par essaim particulaire - faire une optimisation mono-objective et globale
	[CI, 03]	- TRIBES	- paramétrage automatique de l'optimisation par essaim particulaire
	[LJ, 01]	- SIA « Clonalg »	- sélection clonale " source d'inspiration " pour l' optimisation mono-objective et globale - simulation les systèmes immunitaires naturels (mémoire, la reconnaissance de formes, l'apprentissage.....) - effectuer les tâches de filtrage et d'optimisation.

Tableau II.5: Bilan des approches de l'optimisation globale approximative(méta-heuristiques)[MH,13]

C. La sélection hybride

Cette approche est un compromis des deux précédentes approches, en commençant la recherche par une optimisation globale, puis continuant le travail avec une optimisation locale, l'approche peut également manipuler des contraintes globales, les travaux qui prenant en charge cette idée sont résumés en (Tableau II.6).

	Etudes	Outils	Principe de travail
La selection hybride	- [MN, 12]	programmation mixte (MIP)	- résoudre le problème de composition de service web à base QoS : - des variables binaires de décision sont employées dans le modèle pour représenter les candidats de service . - un service candidat s_{ij} est choisi dans la composition optimale si son variable x_{ij} correspondant est placé à 1 dans la solution du modèle et jeté autrement .

Tableau II.6 : Bilan des approches de la sélection hybride[MH,13]

II.4.3.2. La sélection multi-objective

Elle peut être définie comme la recherche de la meilleure ou des meilleures solutions possibles d'un problème donné. Souvent, les problèmes d'optimisation sont des problèmes multi-objectifs. Si les objectifs sont antagonistes, alors on n'a pas une seule solution optimale mais un ensemble de solutions de compromis.

II.4.3.3. La sélection mono et multi-objective

C'est une approche hybride qui combine les deux sélections mono et multi-objective, telle qu'elle effectue une recherche (ou filtrage) multi objective pour chaque classe, après elle regroupe hiérarchiquement les services de chaque classe en choisissant un représentant de ces groupes, ensuite elle continue avec une recherche mono-objective (par niveau) sur les représentants des groupes.

II.5.Problématique

La problématique de sélection de services web basée sur la QoS constituée, si la réponse à la requête d'un client exige la combinaison de plusieurs services existants, alors la sélection dans ce cas, sera plus complexe du fait qu'il faut choisir la combinaison des services composants qui répond mieux aux besoins des clients.

II.6.Conclusion

Dans ce chapitre, on a présenté la sélection de services web et citer les différentes approches de sélection proposées dans la littérature. Ensuite, on a formalisé le problème lié de la sélection automatique de services web.

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

Partie 02

Contribution

Chapitre III

Approche proposée

III.1.Introduction

Dans ce travail, nous proposons une stratégie de sélection des services web pour une requête d'un client et nous utilisons un modèle à base de poids pour un ensemble de paramètres de QoS (les paramètres de QoS sont définis selon les besoins du client). Pour la sélection du meilleur service web, on a besoin d'appliquer l'un des algorithmes d'optimisations qui sont les algorithmes mimétiques.

III.2.Description de la base

- Un service web est un système logiciel identifié par un URI dont les interfaces publiques et les incarnations sont définies et décrites en XML. [W3C;16]
- Une tâche est une opération déterminée par le client à partir les entrés et les sorties.

III.2.1 Description de la base

Notre base des services Web est un fichier XML qui contient (n=120) services web, comme exemple le document ci-dessous :

<p><Services></p> <p>Service 1 [<service id="1" lat="123" cout="15" rep="3"> <inputs> <="" inputs><br="" p1;p2;p3;p5=""></inputs>> <outputs> p4;p6;p9 </outputs> </service></p> <p>Service 2 [<service id="2" lat="250" cout="22" rep="2"> <inputs> <="" inputs><br="" p1;p2;p3;p5=""></inputs>> <outputs> p4;p6;p9 </outputs> </service></p> <p>Service 3 [<service id="3" lat="400" cout="44" rep="12"> <inputs> <="" inputs><br="" p31;p8;p7;p6;p14;p43=""></inputs>> <outputs> p20;p15;p19 </outputs> </service></p> <p>.....</p> <p></Services></p>	<p><Tâches></p> <p>Tâche 1 [<tâche id = "1"> <inputs> <="" inputs><br="" p3;p8;p13;p51=""></inputs>> <outputs> p24;p7;p10 </outputs> </tâche ></p> <p>Tâche 2 [< tâche id = "2"> <inputs> <="" inputs><br="" p1=""></inputs>> <outputs> p5 </outputs> </tâche ></p> <p>Tâche 3 [< tâche id = "3"> <inputs> <="" inputs><br="" p3;p2;p4;p9=""></inputs>> <outputs> p25;p15;p20 </outputs> </tâche ></p> <p>.....</p> <p></ Tâches ></p>
--	--

III.2.2 Description de la requête

La requête de l'utilisateur consiste à vérifier la qualité de services de la composition sélectionnée par le système, en considérant les valeurs minimales ou maximales des trois critères de QoS suivantes :

- La borne minimale de la réputation
- La borne maximale du coût

- La borne maximale de temps latence,

Ses bornes sont fixées par défaut : 30 pour le coût, 300 pour latence et 2 comme minimum de la réputation. Nous considérons aussi des poids égaux pour ces critères de QoS par défaut 30% pour le coût, 30% pour latence et 40% pour la réputation, sachant que ces valeurs peuvent être changées selon les besoins des utilisateurs.

III.3.Algorithme Mimétique

III.3.1 Définition

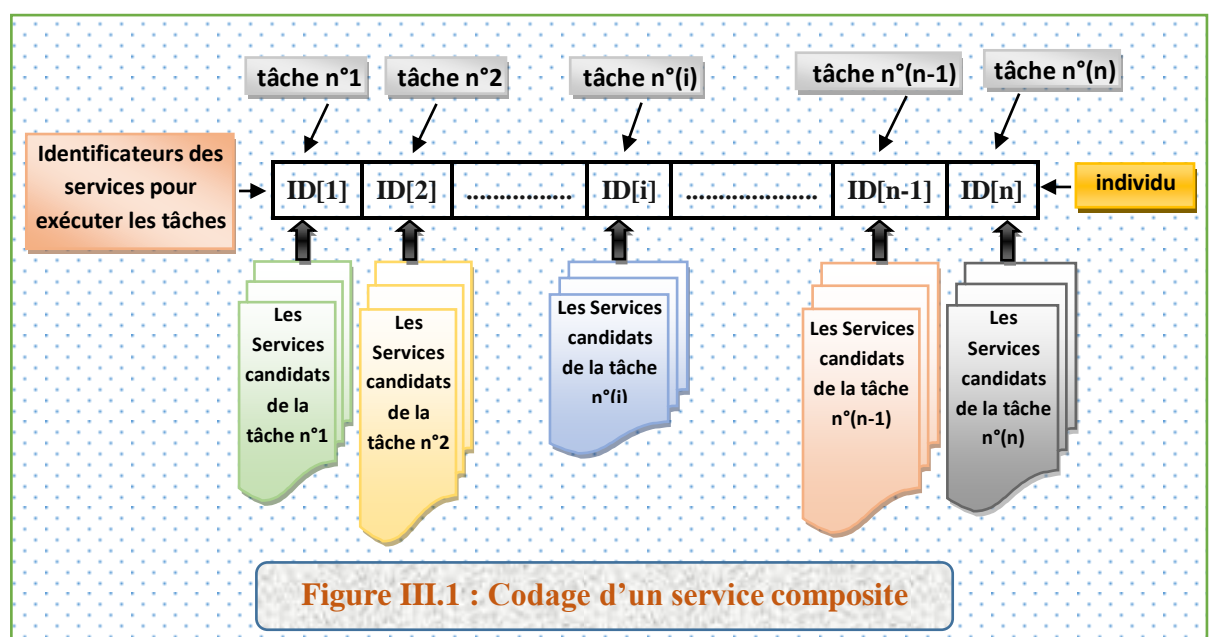
Les algorithmes mimétiques ont été introduits par Dawkins et formalisés par Moscato [MO, 89]. Leur principe général consiste à hybrider un algorithme génétique avec un algorithme de recherche locale, ils sont appelés aussi algorithmes génétiques hybrides, et Recherche locale Hybrides.

III.3.2 Algorithme génétique

III.3.2.1 Codage d'individus

Dans notre approche, on suppose que le service composite est une séquence des tâches, le **chromosome** (l'individu) est défini par un tableau de taille n (où n est le nombre de tâches), chaque case dans ce tableau (gène) représente l'identificateur d'un service web parmi d'un ensemble des services qui exécutent sa tâche. **La Figure III.1** montre le codage d'un service composite avec n tâches.

- **Génération** (appelée aussi population) : m individus (taille n) qui représentent un ensemble des services composites à un instant donné.



III.3.2.2 Fonctions d'évaluation

A. Critères de QoS considérés

Dans notre système, nous focalisons sur les trois critères de QoS suivants pour calculer la qualité de la composition de services Web sélectionné :

1. **Coût** : est défini le montant pour exécuter une requête.

- **Formule cout** = $cout(sol)$
- **Quantification** : monnaie (DA).

2. **Latence "temps de réponse"** : c'est le temps nécessaire pour exécuter une requête

- **Formule latence** ($lat(sol)$) = Le temps de réception de la réponse par le client – Le temps d'envoi de la requête par le client.
- **Quantification** : seconde.

3. **La Réputation** : c'est une mesure de la fiabilité de l'opération.

- **Formule Réputation** = $Rep(sol)$
- **Quantification** : Pourcentage

B. Fonctions d'agrégation

Nous avons besoin de calculer les paramètres de QoS pour la composition des services web, en se basant sur la signification de chaque critère de QoS, nous définissons un ensemble de fonctions d'agrégation permettant de calculer les valeurs agrégées de chaque critère de QoS. (Tableau III.1)

Critères de QoS	Fonctions d'agrégation
Latence	$Lat(Sol) = \sum_{i=1}^n lat(s_i)$
Cout	$Cout(Sol) = \sum_{i=1}^n cout(s_i)$
Réputation	$Rep(Sol) = \frac{1}{n} \sum_{i=1}^n rep(s_i)$

Tableau III.1 : Les fonctions d'agrégation des critères de QoS

C. Fonction objective

La qualité du web service composite peut être décrite par un vecteur de QoS comme suit:

$$\mathbf{QoS(sol)} = \mathbf{1/f(rep(sol), lat(sol), cou(sol))}$$

La latence et le coût prennent des valeurs scalaires dans \mathbf{R}^+ , la réputation varie sur l'intervalle [1,5], voir le tableau (Tableau III.2) .

Critères de QoS	Valeurs
Latence (lat)	1 – 300 (ms)
Coût (cout)	1 – 30 (\$)
Réputation (rep)	1 – 5

Tableau III.2 : Les intervalles des critères de QoS [BD, 13]

✓ La fonction d'évaluation prenant la forme suivant :

$$f(sol) = \frac{1}{|Pr * rep(sol) - Pc * cout(sol) - Pl * lat(sol) |}$$

Pl : le poids de latence définit par l'utilisateur.

Pc : le poids de cout définit par l'utilisateur.

Pr : le poids de réputation définit par l'utilisateur.

III.3.2.3 Opérateurs génétiques

a) Sélection : la sélection des individus les mieux adaptés par leur fonction d'adaptation, qui doivent se reproduire. Cette opération se fait par la stratégie (Sélection par roulette).

b) Croisement : Le croisement permet de créer de nouveaux individus en échangeant des services composants entre deux individus.

- **Taux de croisement**: L'opération de croisement est appliquée avec une probabilité **Pcroi**

c) Mutation: La mutation est exécutée seulement sur un seul individu. Elle représente les services des tâches aléatoires par un service de l'ensemble des services candidats pour chaque tâche mutée.

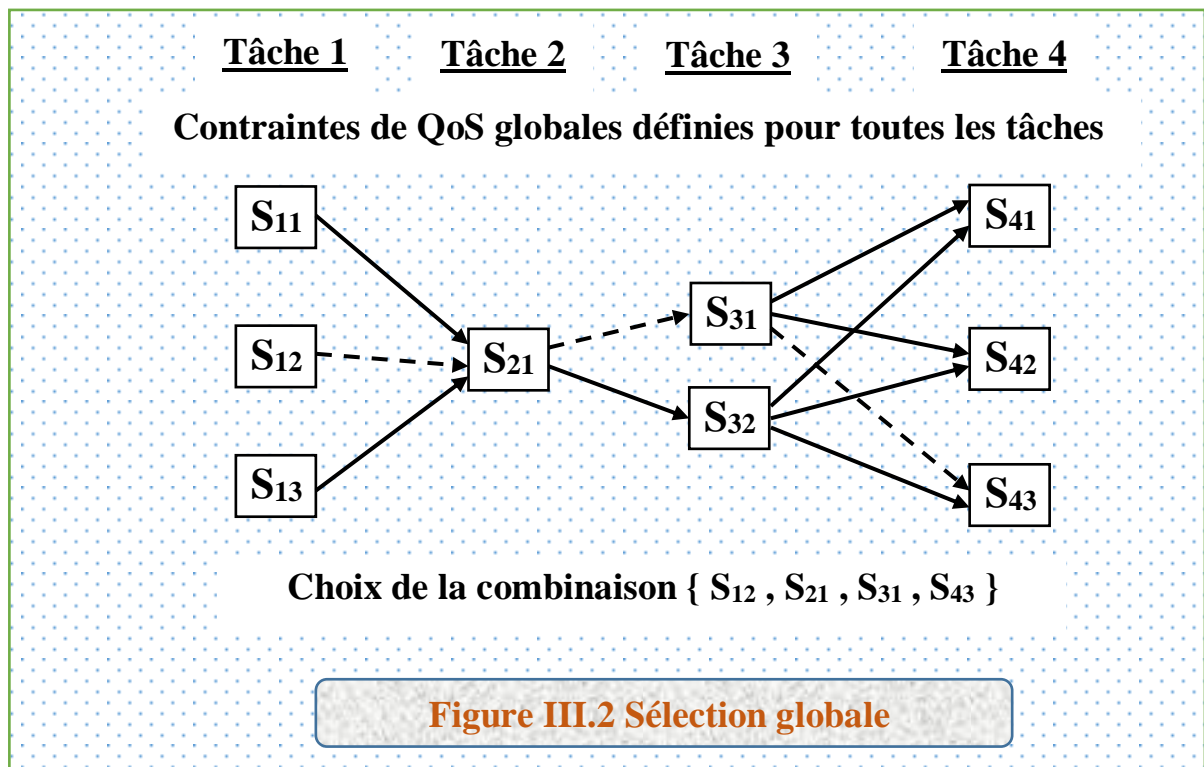
- **Taux de mutation** : L'opération de mutation est appliquée avec une probabilité **Pmut**

d) **Remplacement** : remplacer les mauvaises individus par des autres.

III.3.3 Algorithme de recherche locale

l'ajout de la recherche locale va changer le comportement de l'algorithme :

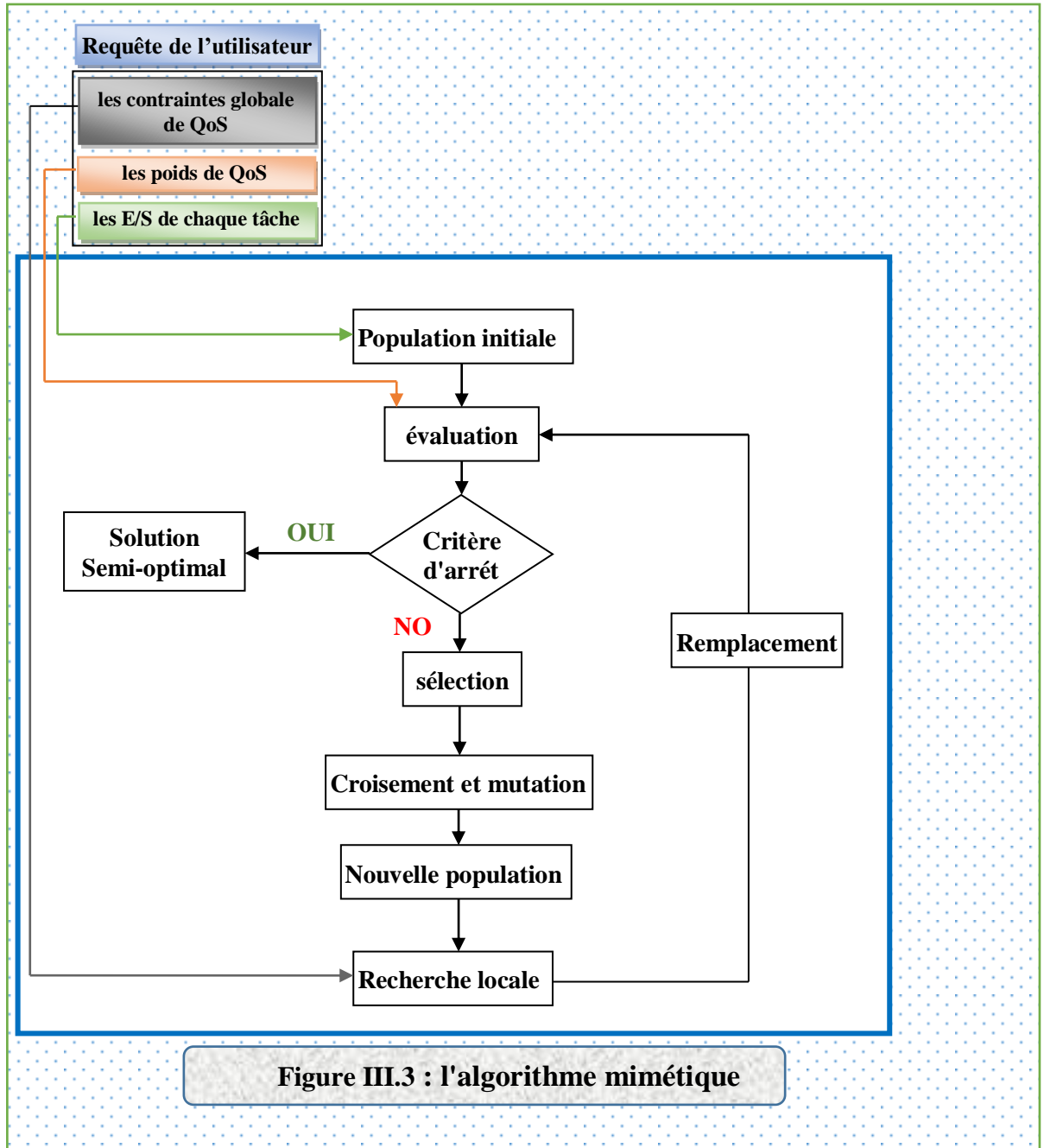
- La première utilisation de la procédure recherche locale permet de travailler avec des solutions bien intensifiées et qui sont parmi les meilleures.
- La deuxième utilisation de la recherche locale permet d'obtenir les meilleures enfants de la nouvelle génération créé.
- ✓ La modélisation de notre problème dépend de la nature de la stratégie de sélection globale, on choisit la combinaison de services web qui garantit la meilleure qualité globale pour l'ensemble des tâches, la **Figure III.1** décrit un exemple de la stratégie de sélection globale.



La procédure de recherche locale dans l'algorithme mimétique s'effectue sur tous les individus de la population initiale, après elle s'applique sur chaque nouvel individu issu des opérations de croisement et mutation, c'est-à-dire qu'on tente toujours de choisir les meilleurs individus de la génération construite.

- ✓ Donc, les deux méthodes (algorithme génétique et algorithme de recherche locale) sont complémentaires, la **Figure III.3** décrit le principe de l'algorithme mimétique.

✓ L'algorithme mimétique



✓ Description de l'algorithme mimétique

- **Etape 1** : construire une population initiale de m individus de manière aléatoire .
- **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. qui permet de créer un nouvel individu à partir de deux individus parents.

- **Etape 4** : appliquer l'opérateur de mutation pour introduire des nouveaux individus dans la population.
- **Etape 5** : appliquer l'opérateur de recherche locale aux nouveaux individus créés (enfants). et renvoie les meilleurs individus trouvés , pendant nombre défini d'itérations.
- **Etape 6** : remplacer les individus les plus mauvais par les nouveaux individus créés en l'étape (5).

III.4.Conclusion

Dans ce chapitre, on a proposé un algorithme à base d'optimisation mimétique pour le problème de sélection de service web à partir de la requête de client, Nous avons présenté les différentes étapes de la conception.

Dans le chapitre suivant, Nous allons présenter les résultats de la validation de notre approche, afin d'évaluer leur efficacité.

Chapitre IV

Implémentation

et

Expérimentations

IV.1.Introduction

Dans le présent chapitre nous décrivons un prototype implémentant cette architecture ainsi que les résultats obtenus.

Nous allons commencer par présenter les différents outils techniques liés à l'implémentation, ensuite nous présentons notre prototype et les expérimentations menées pour analyser notre approche.

IV.2.Présentation des outils technologiques utilisés

Le prototype a été développé sous Windows 7 sur une machine qui contient un processeur Intel i3 de vitesse 3.40 GHZ de 4 GB de RAM , en utilisant des outils open source graphiques et développés en JAVA.

Nous détaillons, dans ce qui suit, chacun des outils et langages utilisés pour la manipulation des données ainsi que l'implémentation de l'interface utilisateur.

✓ Le 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 , enfin il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers , les fonctions multimédia et beaucoup d'autres fonctionnalités;

✓ L'IDE NetBeans :

Nous avons développé notre application sous l'environnement de développement intégré (IDE) NetBeans version 8.0.2 (Build 201411181905) pour Java, sous licence CDDL (common development and Distribution license) version 1.0.

Le choix de NetBeans était fondamental puisqu'il est un logiciel permettant principalement le développement en java et il fournit un environnement convivial en termes d'utilisation.

✓ L'API JDOM :

JDOM est une API open source Java, vue comme un modèle de documents objets dont le but est de représenter et manipuler un document XML de manière intuitive pour un développeur Java sans requérir une connaissance pointue de XML.

IV.3. Implémentation

▪ Présentation de l'Interface Humain/Machine (IHM)

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

Par la suite nous donnons quelques captures d'écran qui montrent les différentes parties, composantes et fenêtres de notre application :

✓ Commenant par la fenêtre principale de l'interface (*figure IV.1*), cette fenêtre donne une idée générale sur le système.

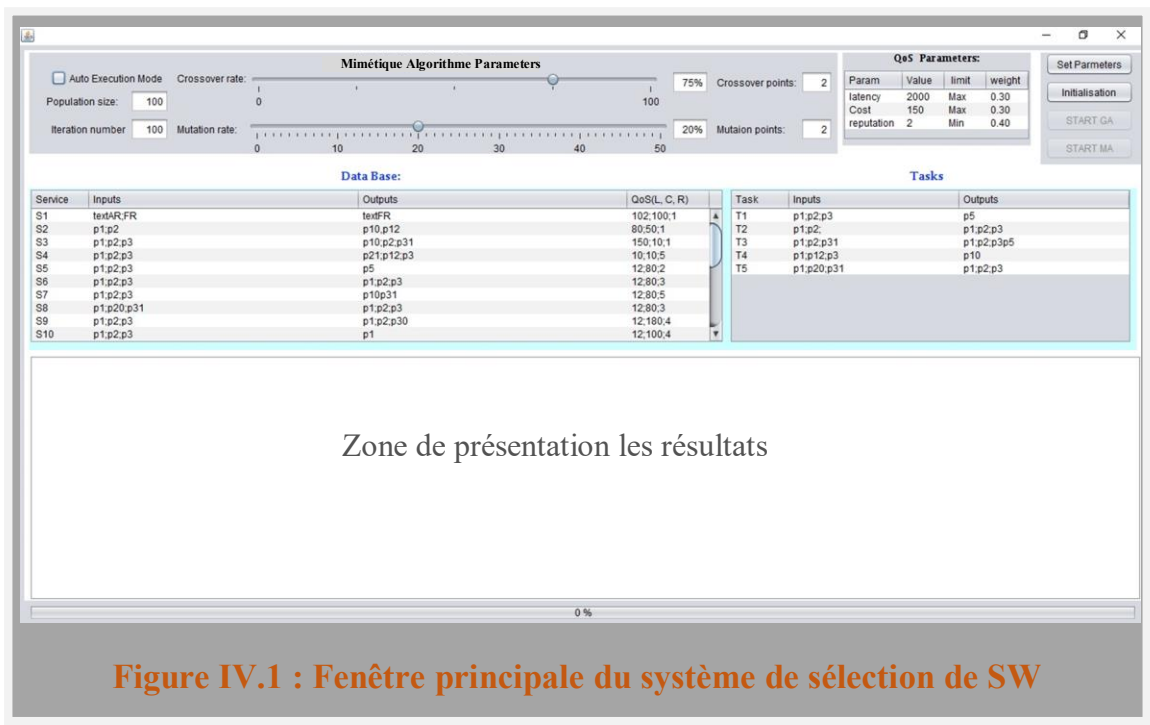


Figure IV.1 : Fenêtre principale du système de sélection de SW

✓ La première section (*figure IV.2*) concerne notre base des services Web.

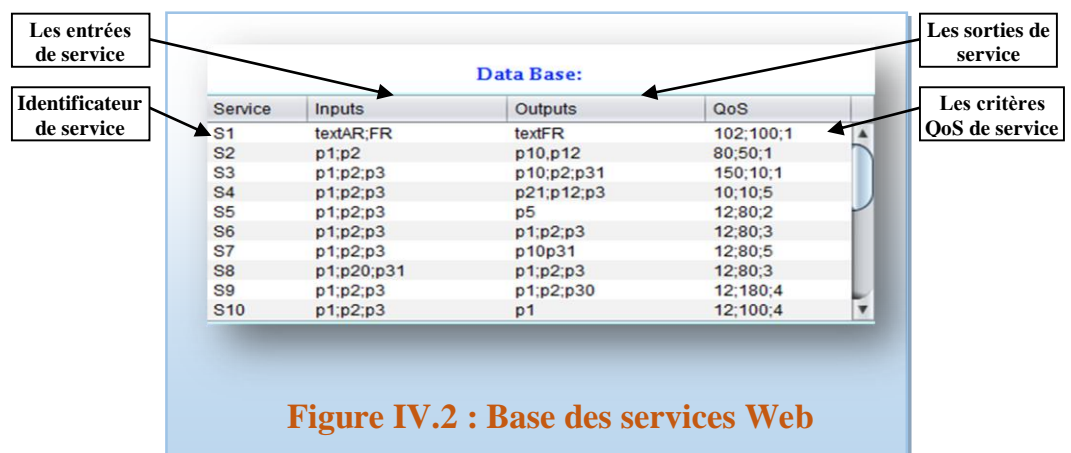
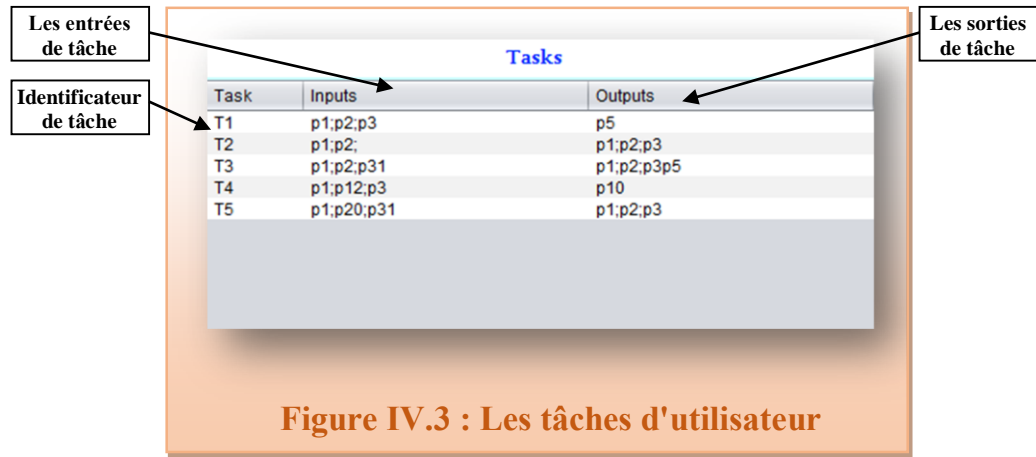
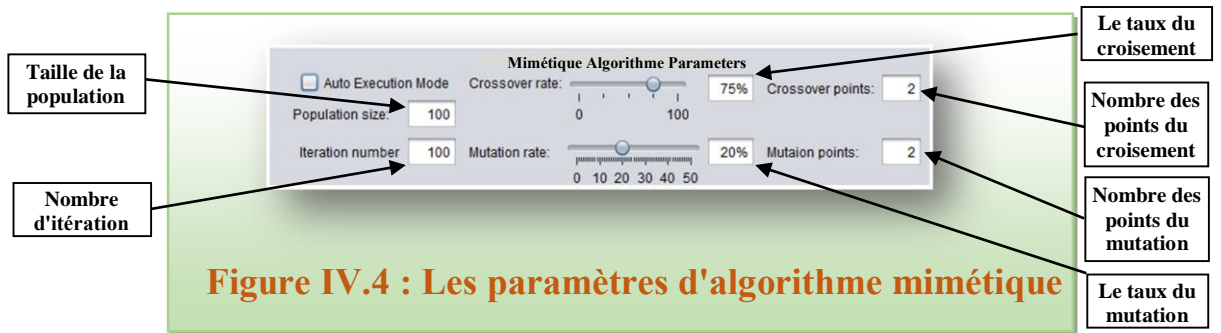


Figure IV.2 : Base des services Web

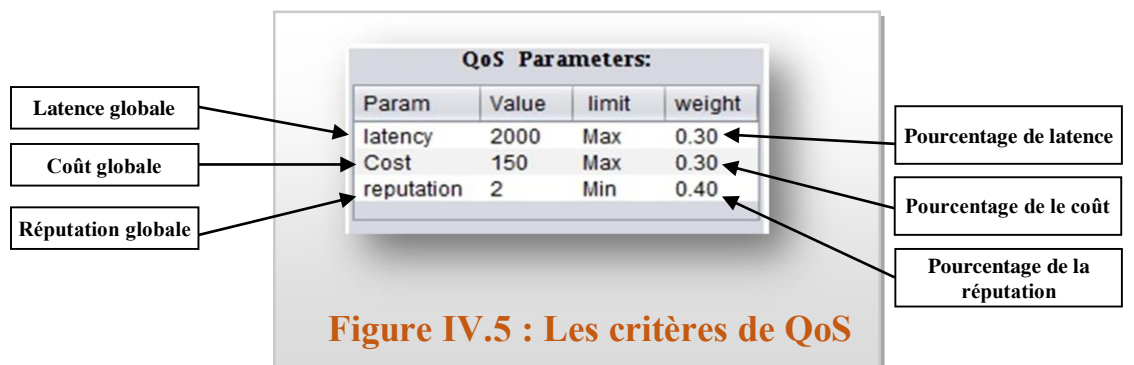
✓ La deuxième section (*figure IV.3*) concerne les tâches qui ont défini par l'utilisateur.



✓ La troisième section (*figure IV.4*) concerne les paramètres qui sont intervenus dans les opérations mimétiques (croisement et mutation).



✓ La quatrième section (*figure IV.5*) concerne les critères de QoS qui sont intervenus dans la recherche locale.



IV.4.3. Troisième configuration

Nous prenons les mêmes paramètres de la 1^{ière} configuration mais le taux de croisement est changé:

	Taux de croisement	La valeur de fonction fitness
1 ^{ière} simulation	30%	0.0244
2 ^{ième} simulation	40%	0.0277
3 ^{ième} simulation	60%	0.0271
4 ^{ième} simulation	70%	0.0263
5 ^{ième} simulation	80%	0.0250

Tableau IV.2 : Résultats de la 3^{ième} configuration

IV.4.4. Quatrième configuration

Nous prenons les mêmes paramètres de la 1^{ière} configuration mais le taux du mutation est changé :

	Taux du mutation	La valeur de fonction fitness
1 ^{ière} simulation	40%	0.0238
2 ^{ième} simulation	50%	0.0227
3 ^{ième} simulation	60%	0.0208
4 ^{ième} simulation	70%	0.0217
5 ^{ième} simulation	80%	0.0209

Tableau IV.3 : Résultats de la 4^{ième} configuration

IV.4.5. Discussion

D'après nos expériences on conclut les points suivants :

- le nombre d'itérations est élevé mieux seront les résultats d'optimalité malgré l'augmentation du temps d'exécution.
- le taux 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 50% dès le début).
- La Même chose pour le taux de mutation , après plusieurs simulations nous avons conclu que la valeur 30% est la meilleure configuration pour le taux de mutation.

IV.4.6. 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 (IV.4)** montre les résultat de cette comparaison pour la deuxième configuration précédente.

	Algorithme mimétique	Algorithme génétique
	Fitness de la solution optimale	Fitness de la solution optimale
<i>1^{ière} simulation (après 200 itérations)</i>	<i>0.0204</i>	<i>0.0205</i>
<i>2^{ième} simulation (après 300 itérations)</i>	<i>0.0263</i>	<i>0.0212</i>
<i>3^{ième} simulation (après 400 itérations)</i>	<i>0.0371</i>	<i>0.0233</i>
<i>4^{ième} simulation (après 500 itérations)</i>	<i>0.0670</i>	<i>0.0285</i>
<i>5^{ième} simulation (après 600 itérations)</i>	<i>0.0840</i>	<i>0.0385</i>
Max	0.0840	0.0385

Tableau IV.4 : résultat de comparaison (algorithme génétique / mimétique)

- ✓ On conclut que les valeurs de la fonction fitness de l'algorithme mimétique sont meilleurs par rapport les valeurs de l'algorithme génétique, ce qui valide l'efficacité de l'algorithme mimétique que l'algorithme génétique

IV.5.Conclusion

Dans ce chapitre on a appliqué 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

Dans ce mémoire, Nous avons présenté les technologies liées à la sélection de services web dans la littérature. Ensuite, nous avons proposé une approche de sélection à base d'un algorithme mimétique à partir d'une requête du client exige un service composite.

Notre prototype de sélection prend de contraintes globales et trois (03) critères de QoS : (un critère positif : Latence et les critères négatifs : coût et réputation). Pour l'optimisation de la sélection de service web, en utilisant une fonction d'évaluation qui permet de choisir les services composants les plus satisfaisantes de service composite.

Ce projet de fin d'étude permis d'instruire et améliorer nos connaissances, Il est considéré comme une référence pour les futurs travaux.

Pour une continuation de notre travail, plusieurs perspectives peuvent être envisagées :

- Comme on peut résoudre ce même problème par une autre méthode de programmation " La programmation par contrainte " ou " La programmation dynamique " .
- Il est possible de faire une optimisation multi objective ou hybride pour résoudre le même problème proposé dont le but d'améliorer les résultats finaux...

Références Bibliographiques

- [BE,12] : Les Algorithmes Mimétiques ; Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique Melle. BELMEBROUK Zineb.
- [BO,12] : Sélection des Web Services Sémantiques Mémoire de Magister En Informatique ; BOUDJELABA Hakim.
- [BD,13] : La sélection de services Web À base de l'algorithme d'abeille Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique Melle BOUROUIS Meriem Et Melle DJERRIRI Saba Juillet 2013.
- [BK,12] : Sélection des services Web à base d'algorithmes génétiques multi-objectif ; Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique Belkhiter Khalid
- [CGS,03] : 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,
- [CNW,01] :F.Curbera, A.Nagy, et S. Weerawarana. "Web-services : Why and how". Dans Workshop on Object-Oriented Web Services (in OOPSLA).
- [GH,12] : ALGORITHME DE SELECTION DANS LES APPLICATIONS A SERVICES ; Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique Ghazi GHARBI.
- [MH,13] : Sélection de Service Web à base d'algorithme Mimétique ; MOULAY KHATIR Oussama Mohy-eddine HENAOUI Redouan
- [MO89] : P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms", Technical Report C3P 826, Cal-teech Concurrent Computation Program.
- [JET,02] : O'Sullivan Justin, Edmond David, Ter Hofstede Arthur, What's in a service Distrib. Parallel Databases.
- [TLH,10] : Etude de Problème de Sélection de Services Web en tenant Compte des contraintes temps réel ; TURKI KOSSSENTINI1Hazar, Dr. Leila BACCOUCHE, Pr. Henda HAJJAMI BEN GHEZALA