République Algérienne Démocratique et Populaire
Ministère de L'enseignement supérieur et de la Recherche scientifique
Université du 20 Aout 1955 Skikda
Faculté des Sciences
Département d'informatique

Mémoire d fin d'étude pour l'obtention du diplôme

Master Informatique

Option

Réseaux et Système Distribués

Thème

Sélection et composition des services web par la recherche tabou

Réalisé par :

- Saoula Ryene
- Foufou fatima zohra

Encadré par :

M. BOUHOUCHE ABDELOUAHID

Dr MAZOUZI Smaine

Année universitaire :2023 -2024

Remerciements

Je voudrais dans un premier temps remercier, mon encadreur de mémoire Dr. BOUHOUCHE, professeur de l'informatique à l'université de 20 aout Skikda, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.

Je remercie mes très chers parents, Saci et Nadia, qui ont toujours été là pour moi. Je remercie mes sœurs et mes frères, pour leurs encouragements. À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Merci

Dédicace

Je dédié ce travail

A toute ma famille.

A tous mes amis.

A tous ceux qui m'aime.

Et tous ceux que j'aime.

RYENE

Dédicace

Je remercie dieu pour sa gras et don aide dans l'accomplissement de ce mémoire

Je me remercie pour les efforts et la préséance que j'ai déployés pour réaliser cet accomplissement

Je remercie 2galement ma mère et ma grand-mère pour leurs prières et leur soutien continu…

Fatima Zohra

Résumé

La croissance constante des services web sur l'internet mondial pose de nouveaux défis pour les secteurs académique et industriel. En effet, la présence de services web similaires sur le plan fonctionnel mais différents en termes de qualité de service (QoS) nous oblige à mettre en œuvre des techniques d'optimisation pour identifier les meilleures compositions de services. Pour résoudre le problème de la sélection des services web, nous proposons dans notre étude une nouvelle fonction objective et l'utilisation de l'algorithme de recherche tabou pour calculer son optimum global. Cet optimum global correspondra à la meilleure sélection de services web répondant aux exigences du client.

Les résultats obtenus avec une application implémentant notre méthode sont très satisfaisants et encouragent de futurs travaux dans ce domaine de recherche.

Mots clés: Architecture orientée service, Composition de services, Sélection de services, Fonction objectif, Qualité de service, Optimisation combinatoire, Méta heuristiques, Recherche tabou.

Abstract

The continuous growth of web services on the global internet presents new challenges for both academic and industrial sectors. Indeed, the presence of similar web services in terms of functionality but differing in terms of quality of service (QoS) requires us to implement optimization techniques to identify the best service compositions. To address the problem of web service selection, our study proposes a new objective function and the use of the tabu search algorithm to calculate its global optimum. This global optimum will correspond to the best selection of web services that meet the client's requirements. The results obtained using an application implementing our method are very satisfactory and encourage future work in this research area.

Keywords:

Service Oriented Architecture, Service Composition, Service Selection, Objective Function, Quality of service, Combinatorial Optimization , Meta Heuristics, taboo Search .

ملخص:

النمو المستمر للخدمات الويب على الإنترنت التعاطي يخلق تحديات جديدة لقطاعين الأكاديمي اختيار والصناعي. في الواقع، وجود خدمات ويب متشابهة من الناحية الوظيفية ولكن مختلفة من حيث جودة لحساب الحد الخدمة (QoS) يلزمنا طبيعة تقنيات تحسين الاختيار أفضل تكوينات للخدمات. لحل مشكلة التي تلبي متطلبات خدمات الويب نقترح في دراستنا وظيفة هدف جديدة واستخدام خوارزمية البحث المحظور العميل. الأقصى الإعطاء لها. هذا الحد الأقصى الإعطاء بمثل أفضل اختيار خدمات للويب

المجال. – النتائج التي تم الحصول عليها باستخدام تطبيقية فريدة طريقة كانت مرضية للغاية وتشجع على التطبيق أعمال بحثية مستقبلية في هذا

الكلمات المفتاحية:

هندسة موجهة بالخدمات، تكوير الخدمات، اختيار الخدمات، وظيفة الهدف، جودة الخدمة، تحسين التوليفات الميتا-استراتيجيات البحث المحظور.

Table des matières :

Liste Des Figurines:

Liste Des Tableaux:

Glossaire:

WSDL: Web Services Description Langage.

SOAP: Simple Object Access Protocol.

HTTP: Hyper Text Transfert Protocol.

XML : eXtensibleMarkup Langage.

W3C: World Wide Web Consortium.

RFQ: Request For Quotation.

API : Application Programming Interface

UDDI: Universal Description, Discovery and Integration.

QoS : Qualité de Service.

SGML: Standard Generalized Markup Language.

HTML: Hyper Text Markup Language.

XSD: XML Schéma Définition.

RMI: Remote Method Invocation.

CORBA: Common Gateway Interface.

DCOM: Distributed Component Object Model.

ACO : AntColonyOptimization.

BDD : Base Des Données

EDI : Environnement de Développement Intégré.

IBM: International Business Machines.

Max: Maximum.

MCDM: Multiple Criteria Decision Making.

Min : Minimum.

Neg : Négatif.

Pos : Positif.

PSO : ParticleSwarmOptimization.

RAD: Rapide Application Développement.

Rest: Representation State Transfers.

RPC: Remote Procedure Call.

SOA: Service Oriented Architecture.

SSW : Sélection des Services Web.

SWC: Service Web Composite.

UML: Unified Modeling Language.

URL: Uniform locator.

Introduction Générale

Introduction Générale:

Le web est devenu une source d'information dynamique où l'information est produite à la demande. Dans ce contexte, les services web sont devenus un atout majeur pour les entreprises modernes.

Les services web permettent la communication entre applications distantes via un réseau (internet ou intranet). Lorsqu'il existe plusieurs services web similaires du point de vue fonctionnel (même comportement ou interface, c'est-à-dire les mêmes entrées et sorties), les utilisateurs doivent choisir en se basant sur des critères de qualité de service (QOS). Ces critères deviennent particulièrement importants lorsque les besoins des clients sont complexes et nécessitent des compositions de services.

2. Problématique:

Avec le nombre croissant de services disponibles sur internet ayant des fonctionnalités similaires, la question de "comment distinguer et sélectionner le meilleur service Web parmi les autres" devient un problème majeur à résoudre.

La plupart des approches existantes dans la littérature se basent sur la sélection des services web selon leur description fonctionnelle. Cependant, ces services web ne peuvent pas toujours garantir une qualité de service adéquate.

La sélection des services web selon les aspects non fonctionnels repose essentiellement sur la qualité de service (QoS), qui se mesure à l'aide de plusieurs métriques : coût, temps d'exécution, fiabilité, disponibilité et réputation.

L'objectif de notre travail est de sélectionner de manière optimale les services web qui répondent aux exigences non fonctionnelles des clients.

Contribution:

Pour sélectionner le meilleur service parmi plusieurs équivalents, nous avons proposé une approche de sélection basée sur une méthode méta-heuristique. Après une étude détaillée, nous avons opté pour la recherche tabou , dont les résultats se sont avérés plus satisfaisants par rapport aux autres approches. De plus, nous avons proposé une nouvelle fonction objectif à minimiser par la reherche tabou.

Plan du mémoire:

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

Chapitre I:

Il traite des concepts essentiels des services web, en abordant l'architecture SOA, l'importance des services web, leurs technologies et leurs standards de base, ainsi que leur fonctionnement et leur architecture. Nous y examinons également leurs avantages et inconvénients, et introduisons les notions de qualité de service (QoS en anglais : Quality of Service).

Chapitre II:

- **Première partie :** Nous introduisons la problématique liée à la sélection des services web.
- **Deuxième partie** : Nous décrivons l'optimisation et ses différents types, illustrés par un exemple concret pour mieux comprendre la motivation derrière cette approche.

Chapitre III:

Le troisième chapitre de la thèse traite de la recherche tabou, une méta-heuristique d’optimisation visant à trouver les meilleurs solutions aux problèmes combinatoires. Elle utilise des techniques de recherche locale et principes fondamentaux de la recherche tabou, les étapes de sa mise en œuvre, ainsi que ses applications pratiques dans des domaines tels que la problème du voyageur de commerce, la planification et la gestion des réseaux.

Chapitre IV:

Dans le chapitre IV, nous avons présenté notre méthode en deux parties, La première partie formule le problème à résoudre et exprime la fonction objective basées sur les critères de qualité de service (Qos), avec l'introduction de recherche tabou, La second présente la conception du système à l'aide de diagrammes UML nécessaires pour notre cas.

Chapitre V :

Ce chapitre est consacré à l'implémentation et la réalisation de l'application qui permet la sélection multi objectif des services web à base d'algorithme recherche tabou.

Chapitre I
Les services Web

1. Introduction:

Depuis les années 70, un besoin de communication entre les systèmes d'information des entreprises (fournisseurs ou clients) a émergé, ce qui a donné naissance à ce que l'on appelle l'intégration B2B. Sur le plan technologique, l'intégration B2B fait référence à "la technologie logicielle utilisée pour connecter n'importe quel système d'information d'une entreprise à tous ses partenaires commerciaux" (C, 2003). Les protocoles qui répondent à ces besoins sont appelés protocoles B2B, dont des exemples incluent l'EDI (http//WWW.x12/org/) et RossetaNet (Http://www/rosettanet.org) .

Avec l'essor du Web ces dernières années, est apparu le besoin de permettre à une application cliente d'invoquer un service d'une application serveur via Internet. Ce besoin a donné naissance à ce que l'on connaît sous le nom de services Web. Étant donné que les services Web permettent de connecter différentes applications, leur utilité devient évidente et importante. Par conséquent, les activités de recherche et développement dans le domaine des services Web sont très dynamiques. Dans ce chapitre, nous décrirons les technologies associées aux services Web.

Définition d'un service web :

On retrouve plusieurs définitions des services Web :

2.1.IBM (International Business Machines Corporation)

Un service Web est une fonction logicielle interopérable qui permet la communication entre machines et est accessible via un emplacement réseau spécifique. Il est conçu avec une interface qui dissimule les détails de son implémentation, ce qui lui permet d'être utilisé indépendamment de la plateforme matérielle ou logicielle et du langage de programmation utilisé. Cette caractéristique favorise une mise en œuvre souple, orientée composants et multi-technologies des applications basées sur les services Web. Ces services peuvent

être utilisés de manière autonome ou combinés avec d'autres pour effectuer des agrégations complexes ou des transactions commerciale

2.2.W3C(World Wide Web Consortium):

Un service Web, défini comme un système logiciel conçu pour permettre des interactions entre machines sur un réseau, repose sur une interface décrite dans un format exploitable par des machines, notamment le langage WSDL. Les interactions avec ce service sont régies par sa description, généralement effectuées à l'aide de messages conformes au protocole SOAP, qui sont transmis via les protocoles Internet tels que HTTP, et utilisent la sérialisation XML ainsi que d'autres normes associées au Web.

3. L'intérêt des services web:

Les services Web offrent de nombreux avantages. Ils peuvent être utilisés à distance sur différentes plateformes, ce qui les rend très polyvalents. Ils sont également utiles pour le développement d'applications distribuées, permettant une collaboration transparente entre différentes parties. De plus, leur accessibilité depuis divers types de clients facilite leur utilisation. En somme, les services Web permettent aux applications de coopérer harmonieusement, offrant ainsi une expérience utilisateur transparente.

3. Caractéristiques des services web:

Les services Web ont les caractéristiques comportementales spéciales suivantes (Http://www.tutprialspoint.com):

-Un service Web est une application logicielle identifiable par un URI (Uniform Resource Identifier).

L'URI est un moyen de localiser un contenu sur le Web, tel qu'un document texte, audio ou vidéo. Par exemple, l'URI le plus courant est l'adresse d'une page

Web. Ainsi, un service Web est accessible en spécifiant son URI, ce qui signifie qu'il est caractérisé par un seul objet et une seule fonctionnalité.

À partir de cette caractéristique, il est possible de construire une application logicielle très étendue comportant plusieurs fonctionnalités. En utilisant des URI spécifiques, il devient alors possible de sélectionner les fonctionnalités désirées au sein de cette application.

-Basés sur XML : Les services Web sont fondés sur XML, tant au niveau de la représentation que du transport des données. L'adoption de XML permet d'éliminer les contraintes liées aux différences de réseau, de système d'exploitation ou de plateforme. Cette utilisation de XML garantit une interopérabilité élevée au niveau de base des applications basées sur les services Web.

-Faiblement couplés : Dans un système faiblement couplé, le lien entre le consommateur d'un service Web et le service lui-même est indirect. Ainsi, une évolution de l'interface du service Web au fil du temps n'affecte pas la capacité du client à interagir avec celui-ci. En revanche, dans un système étroitement couplé, les logiques du client et du serveur sont intimement liées, ce qui signifie qu'un changement dans une interface nécessite une mise à jour de l'autre. Opter pour une architecture faiblement couplée tend à simplifier la gestion des systèmes logiciels et facilite leur intégration avec d'autres systèmes.

-Capacité à être synchrone ou asynchrone : La synchronisation se réfère à la coordination entre le client et l'exécution du service. Dans les appels synchrones, le client bloque son exécution et attend que le service termine son opération avant de poursuivre. À l'inverse, les opérations asynchrones permettent au client d'appeler un service, puis de continuer à exécuter d'autres fonctions. Les clients asynchrones récupèrent le résultat de leur appel plus tard, tandis que les clients synchrones reçoivent leur résultat dès que le service est terminé. La capacité

asynchrone est un élément essentiel pour rendre possible les systèmes à couplage lâche.

- Prend en charge l'échange de documents : L'un des principaux avantages de XML réside dans sa capacité générique à représenter non seulement des données, mais également des documents complexes. Ces documents peuvent varier en complexité, allant de la simple représentation d'une adresse actuelle à des structures beaucoup plus élaborées, telles que la représentation d'un livre complet ou d'une demande de devis (RFQ). Les services Web exploitent cette flexibilité en prenant en charge l'échange transparent de documents, ce qui facilite grandement l'intégration des entreprises en permettant la communication fluide entre différents systèmes et applications.

5. Applications des services web:

Les technologies des services web peuvent être appliquées à diverses applications, offrant des avantages significatifs par rapport aux anciennes API propriétaires, aux implémentations spécifiques à une plate-forme, et à d'autres restrictions classiques (telles que la multi-plateforme, le multi-langage, la disponibilité sur Internet avec des données actualisées en temps réel, etc.).

Ces applications des services web sont variées, couvrant à la fois les domaines B2C (Business to Consumer) et B2B (Business to Business), ainsi que des domaines de gestion tels que la gestion de stock et la gestion commerciale.

Dans le domaine B2C, les services web sont utilisés pour des applications et des sites Internet destinés au grand public. Dans le domaine B2B, ils sont employés pour des applications et des sites Internet destinés au commerce entre professionnels.

Les services web peuvent être utiles dans la plupart des scénarios applicatifs où une communication bidirectionnelle (requête/réponse) peut être

établie. Cependant, leur utilisation n'est pas limitée à cela ; de nombreux autres modèles peuvent également faire appel aux services web, parfois sans que les utilisateurs en aient conscience.

Les entreprises qui fournissent des services web permettent aux développeurs d'utiliser leurs fonctionnalités sans avoir à les recoder, favorisant ainsi le partage des fonctionnalités et simplifiant le processus de développement.

6. Architecture des services web :

6.1. Architecture de référence :

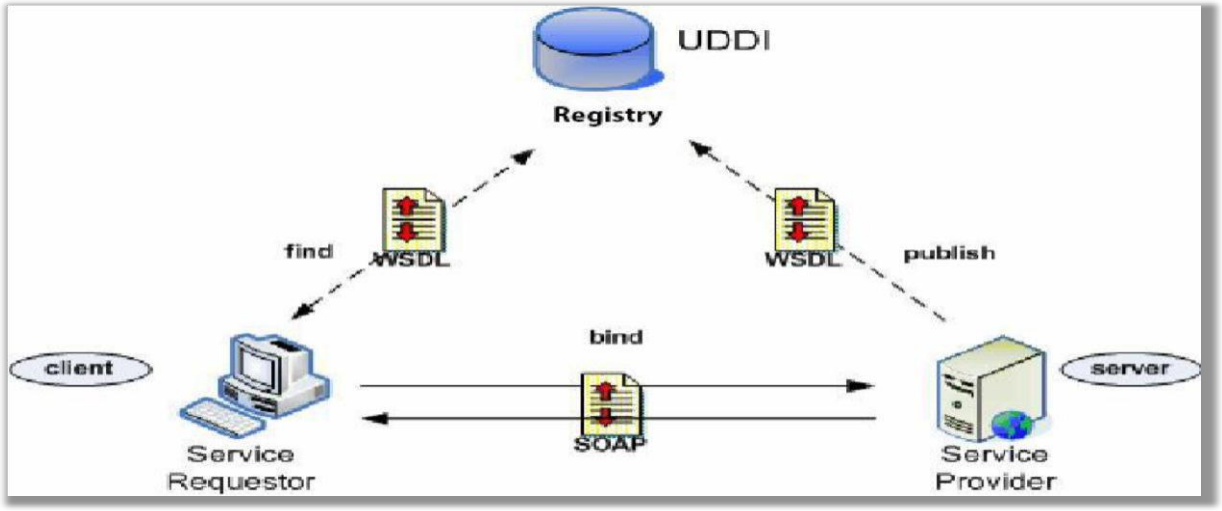
Pour favoriser l'interopérabilité et l'extensibilité du paradigme des services web, une architecture de base est essentielle pour maintenir les objectifs initiaux face aux évolutions technologiques successives.

L'architecture orientée services, comme décrite dans (http://guru99.com), est un modèle architectural dans lequel les composants d'application fournissent des services à d'autres composants via un protocole de communication, généralement sur un réseau.

Cette architecture repose sur trois rôles principaux :

- **Fournisseur de service :** celui qui crée le service web et le publie dans l'annuaire des services, exposant ainsi toutes ses fonctionnalités.
- **Client :** le consommateur qui accède à l'annuaire pour rechercher le service web dont il a besoin, en spécifiant les normes à respecter.
- **Annuaire :** l'entité logicielle qui fait le lien entre les consommateurs et les fournisseurs de services. Elle met à disposition les interfaces d'accès aux services et fournit les contrats et l'architecture nécessaires pour permettre les interactions.

Les interactions de base entre ces trois rôles sont illustrées dans la figure I.1.



FigureI.1 : Architecture de référence [9].

-Publication (publish) : Un fournisseur signale (http://guru99.com) à l'annuaire (registre des services) la disponibilité d'un service Web en utilisant l'interface de publication, permettant ainsi aux clients d'accéder au service.

-Recherche (find) : Un utilisateur consulte l'annuaire (le registre) pour trouver un service Web spécifique qui a été publié.

-Invocation (bind) : En utilisant les informations (description) obtenues de l'annuaire (registre de services) concernant le service Web recherché, l'utilisateur peut alors invoquer ou appeler ce service.

Les services Web sont soutenus par un ensemble de normes de protocoles appelé Web Services Protocol Stack, qui comprend notamment :

- **SOAP :** Connue comme un protocole de messagerie indépendant du transport, SOAP est basé sur le transfert de données XML sous forme de messages SOAP.

- **WSDL:** Il s'agit d'un standard décrivant un service Web sous forme d'une description XML (fichier). Il précise l'emplacement du service ainsi que les méthodes disponibles.
- **UDDI** : Cette norme, basée sur XML, est utilisée pour décrire, publier et rechercher des services Web.

6.2. Architecture en couches:

Une architecture étendue (Http;//www/irit.fr/journali3/hors), souvent appelée pile de services Web, est composée de plusieurs couches superposées les unes sur les autres. La figure 1.2 fournit un exemple de cette organisation en couches. Chaque couche repose sur un standard spécifique, et au-dessus de la couche de transport se trouvent les trois couches qui forment l'infrastructure de base mentionnée précédemment, exploitant les normes émergentes telles que SOAP, WSDL et UDDI.

Comme évoqué précédemment, cette infrastructure de base établit les bases techniques pour rendre les processus métier accessibles non seulement à l'intérieur d'une entreprise, mais également au-delà de ses frontières. Dans ce contexte, deux types de couches contribuent à son enrichissement :

- **Les couches dites transversales :** (Sécurité, administration, transactions et qualité de services (QoS8) rendent viable l'utilisation effective des Web services dans le monde industriel.

- **Une couche Business processus :** permet l'utilisation effective des Web services dans le domaine du e-business.

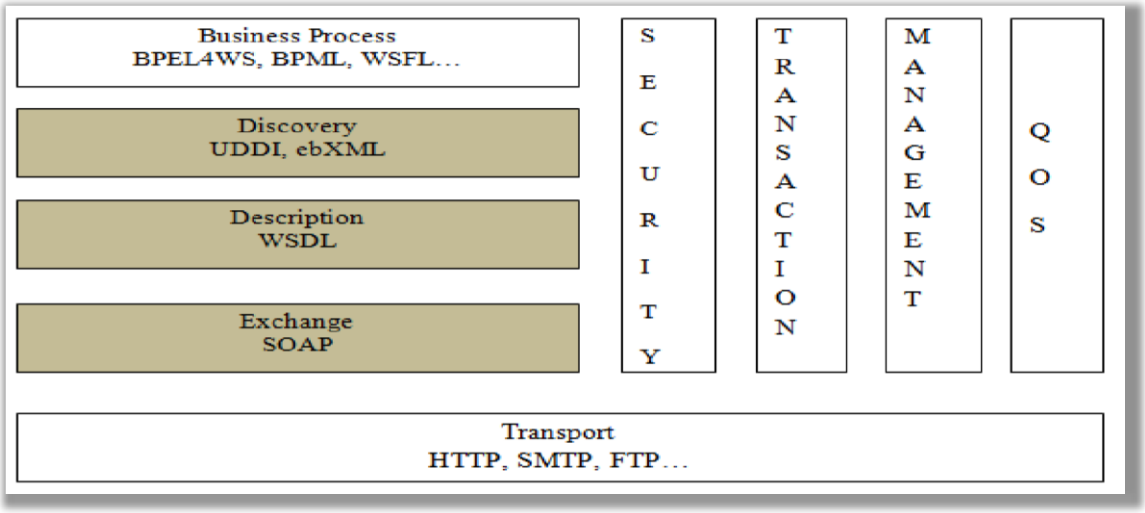


Figure I.2 :Architecture en couches [14].

7.Les principales technologies de développement des services web:

Les services Web emploient un ensemble de technologies qui ont été conçues afin de respecter une structure en couches sans être dépendante de façon excessive de la pile des protocoles.

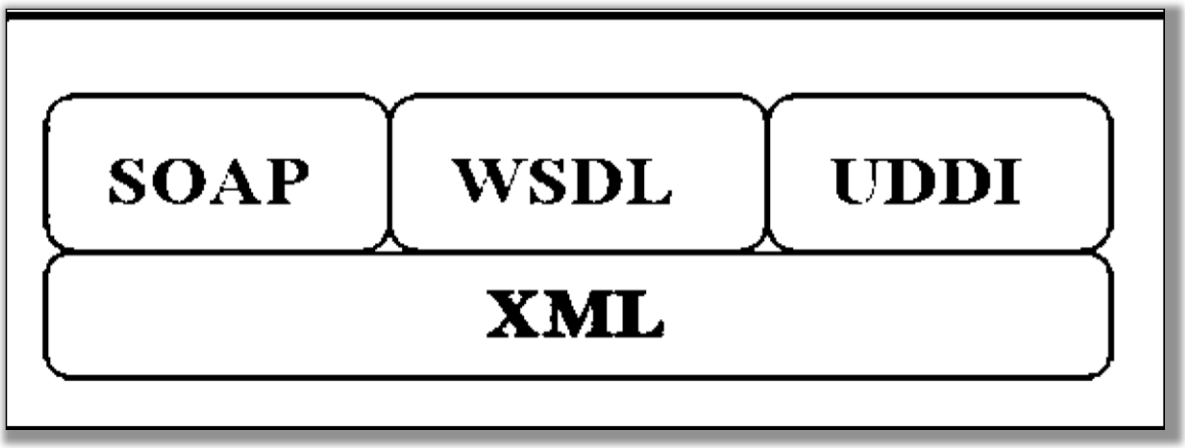


Figure .3 : Architecture des composantes des services web.

On présente les technologies utilisées pour les services web illustré dans la FigureI.3 Comme suit :

7.1. XML (extensible Mark up Language):

XML (http;//www.w3.org) est un ensemble de règles de formatage utilisé pour composer des messages valides. Conçu par le XML Working Group sous l'égide du World Wide Web Consortium (W3C) en 1996, XML est un langage de balisage extensible. Il est apparu comme une réponse à la nécessité de rendre le Standard Generalized Markup Language (SGML9) utilisable sur le Web.

La première spécification officielle de XML a été publiée en février 1998. Contrairement à HTML10, qui se concentre sur la présentation, XML est axé sur les données. Il a permis de transformer Internet d'un espace principalement dédié à l'information et à la présentation de sites web statiques en un environnement web dynamique et programmable, centré sur les données.

XML est conçu pour être lisible par les humains tout en étant interprétable par les machines. Sa flexibilité réside dans sa capacité à définir d'autres langages à partir de ses spécifications de base. En étant indépendant des plates-formes informatiques, XML a ouvert la voie à l'échange et au partage de données entre différentes applications et systèmes.

- **XML_RPC**

XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC (Remote Procedure Call). Dans ce protocole, les requêtes sont rédigées en XML et envoyées via HTTP POST. Les réponses sont ensuite encapsulées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plate-forme, ce qui signifie qu'il peut communiquer avec diverses applications, facilitant ainsi l'intégration entre différents systèmes informatiques.

7.2. SOAP (Simple Object Access Protocol):

SOAP (http;//www.w3.org) est une spécification de communication entre services web qui repose sur l'échange de messages en XML à travers le web.

Cette technologie est simple et facile à implémenter, que ce soit dans des serveurs web ou dans des serveurs d'application. Elle est également indépendante des langages de programmation ou des systèmes d'exploitation utilisés pour l'implémentation des services web.

SOAP fonctionne comme un protocole de communication entre applications, basé sur XML, avec pour objectif principal de faciliter deux types de communications (Chauvest):

1. Servir de protocole de communication au sein des intranets, en vue d'intégrer différentes applications d'entreprise.
2. Faciliter la communication entre applications et services web, en particulier dans le contexte des échanges entre entreprises, comme sur le réseau internet.

- **Structure d'un 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.4) (N.Mitra & Y.Lafon, 2003).

Figure I.4 : Structure de message SOAP.

a) L'en-tête du protocole de transport : qui dépend du 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 : constitue la partie principale d'un message SOAP et est symbolisée par la balise <envelope>. Elle est subdivisée en deux parties distinctes : l'en-tête (Header) et le corps du message (Body).

- L'en-tête du message SOAP est facultatif et extensible. Il est représenté par les balises XML env:Header et </env:Header>. Ces balises peuvent être accompagnées d'attributs pour définir le domaine de noms du service Web.

- En réalité, l'en-tête permet principalement d'ajouter des informations sur le comportement des différents nœuds intermédiaires lors du traitement du message. Un nœud, qui est un intermédiaire SOAP, inclut à la fois le récepteur et l'émetteur SOAP, et est identifiable à partir d'un message SOAP. Son rôle consiste à traiter l'en-tête, puis à transférer le résultat (le message SOAP modifié) à un autre intermédiaire, qui peut être le récepteur final.

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

- **env:role :** Cet attribut est utilisé pour spécifier à quel nœud du réseau le traitement décrit dans le message SOAP est destiné.
- **env:mustUnderstand** : Il s'agit d'un attribut booléen qui indique si le traitement du message est essentiel pour un nœud intermédiaire. Par exemple, cela pourrait signifier qu'un calcul complexe doit être effectué à chaque étape.
- **env:relay** : Cet attribut permet de rediriger un message vers un autre nœud si le premier n'est pas en mesure de le traiter.

- Le contenu du message SOAP, représenté par l'élément **<env:Body>**, contient des données propres à l'application. Ces données sont encadrées par les balises **<env:Body>** et **</env:Body>**, et doivent être sérialisées selon la syntaxe XML. En outre, les données de cette partie peuvent inclure des types spéciaux tels que les messages d'erreur, aussi appelés SOAP Fault.

➢ **Fonctionnement**

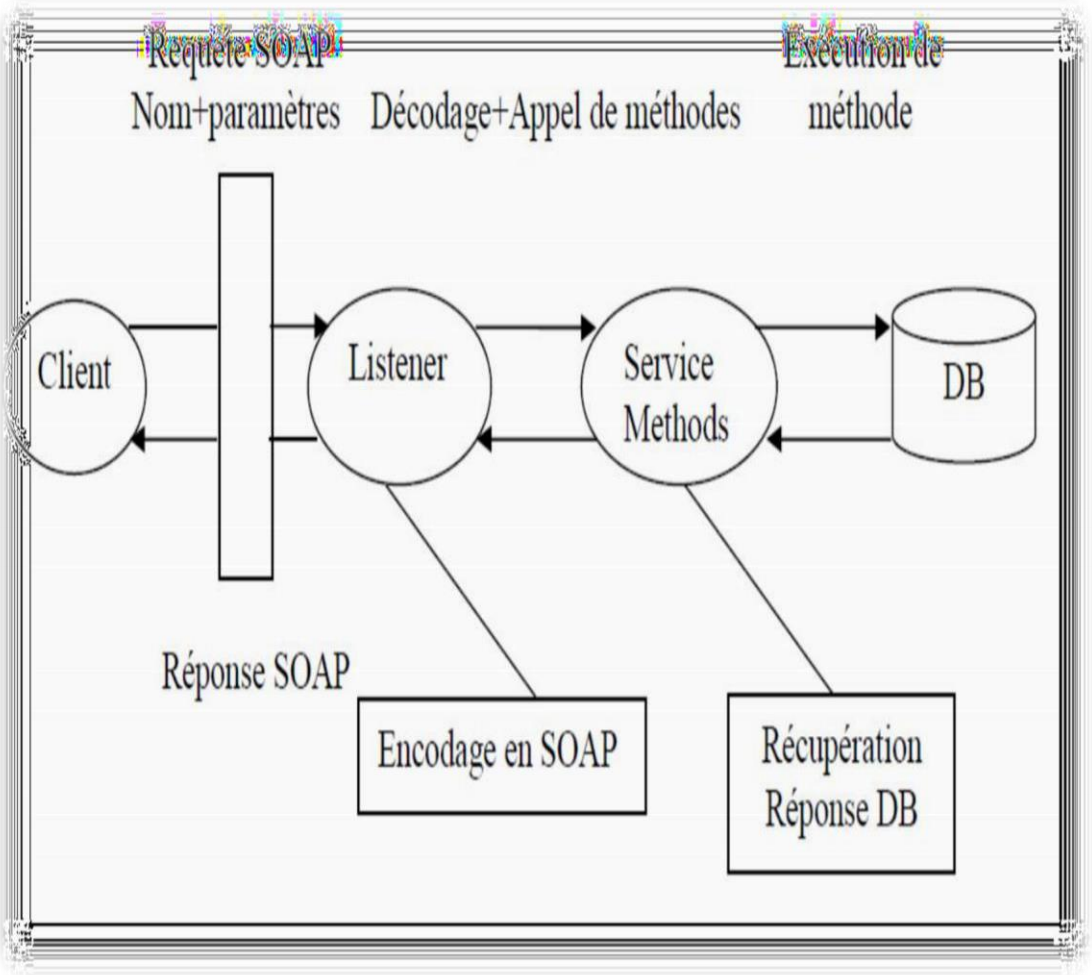


Figure 1.5 : Traitement d'un message SOAP

Lorsqu'un client envoie une requête SOAP via des protocoles de transmission comme HTTP, et que les méthodes requièrent des données provenant de la base de données, le serveur doit se connecter à la base de données. Cela permet au serveur de récupérer ou de manipuler les données nécessaires pour traiter la requête. Une fois les opérations sur la base de données terminées, la connexion est fermée pour maintenir l'intégrité du système et libérer les ressources.

7.3. WSDL (Web Service Description Language):

WSDL (http;//www.w3.org), développé conjointement par IBM, Microsoft et Ariba et adopté par le W3C, offre une norme pour décrire et publier

le format et les protocoles d'un service Web de manière cohérente à travers l'utilisation du format XML. Son utilité réside dans sa capacité à permettre aux requérants et aux fournisseurs de services de comprendre les données qui seront échangées. En effet, il agit comme un langage de définition d'interface (IDL), similaire à CORBA IDL ou aux interfaces de Java, fournissant ainsi une spécification claire des services offertspar un service Web (http://www.w3schools.com).

➢ **Structure et description WSDL :**

WSDL est composé de trois éléments principaux qui peuvent être dissociés et utilisés individuellement ou combinés pour créer un seul document XML. Ces éléments se subdivisent ensuite en sept types de composants descriptifs pour définir les services réseau. Ces éléments sont: (figure I.5) (Leblanc, 14 Novombre 2002) .

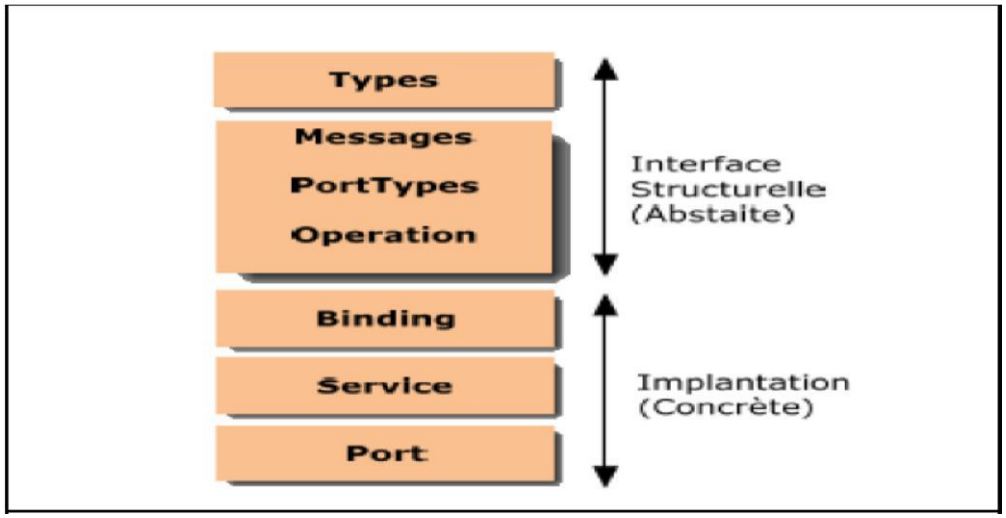


Figure I.6 : Structure et description WSDL.

-Data Type Définition : Cette partie identifie le contenu ainsi que le type de données présentes dans le message.

-Data types : Il s'agit d'une enveloppe pour les définitions de types de données« data type définition », utilisant des systèmes tels queXSD11 (XML Schema Definition).

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

-Abstract Operations : Définit la manière dont les données seront échangées.

-Operation : Description abstraite d'une action prise en charge par le service.

Port Types : Ensemble d'opérations pris en charge par un ou plusieurs points d'accès (ports).

-Binding : Spécification d'un protocole spécifique et d'un format de données pour un point d'accès particulier (port).

-Service Binding : Définit la couche de transport utilisée pour les messages.

-Port : Point d'accès unique défini par une combinaison d'adresse réseau et de numéro de port.

-Service : Ensemble de terminaisons interconnectées.

7.4. UDDI (Universal Description, Discovery and Integration):

UDDI, un annuaire de services, fournit l'infrastructure fondamentale pour la publication et la découverte des services Web. Il permet aux fournisseurs de présenter leurs services Web aux clients. Les informations qu'il contient peuvent être divisées en trois types :

-Pages blanches : Elles comprennent l'adresse, les coordonnées de contact et les identifiants associés au service Web.

-Pages jaunes : Elles identifient les secteurs d'activité liés au service Web.

-Pages vertes : Elles fournissent des informations techniques sur le service.

➢ **Structure de données de l'annuaire UDDI**:

D'après Arnaud (A.Vezain, Février 2005), un registre UDDI se compose de quatre types de structures de données : le BusinessEntity, le BusinessService, le BindingTemplate et le TModel (figure I.6) (A.Vezain, Février 2005) .

Cette classification par type permet une organisation claire facilitant la localisation rapide et la compréhension des différentes informations présentes dans un enregistrement UDDI.

-BusinessEntity : Ces entités contiennent des informations sur les entreprises qui fournissent des services. La recherche peut être effectuée en utilisant les contacts, les noms et les adresses des fournisseurs.

-BusinessService : Ces entités présentent les services en fonction de leurs fonctionnalités, en suivant une taxonomie industrielle standard. Les services peuvent être recherchés par sujet et par domaine.

-BindingTemplate : Ces entités fournissent des informations techniques sur les services Web, ainsi que des références aux descriptions WSDL. La recherche de services se fait en fonction de leurs caractéristiques techniques.

-TModel : Cet ensemble d'informations concerne le mode d'accès aux services. Il peut s'agir également d'une spécification abstraite ou d'une taxonomie (M.Vialette, 8 mars 2006).Haut du formulaire

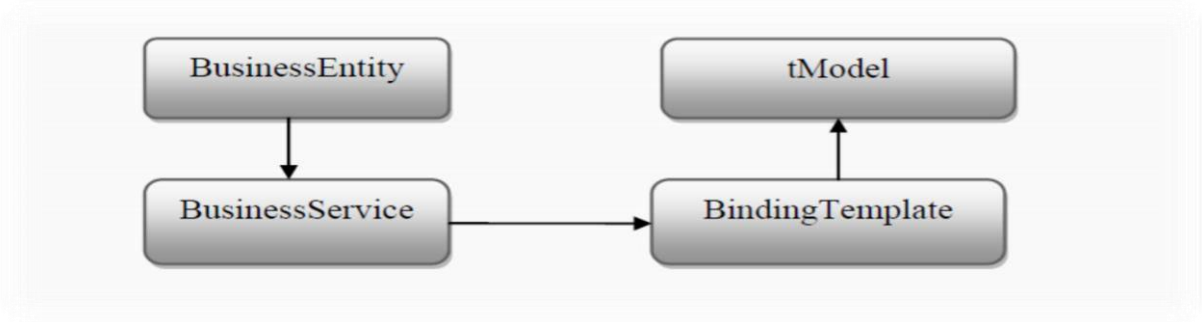
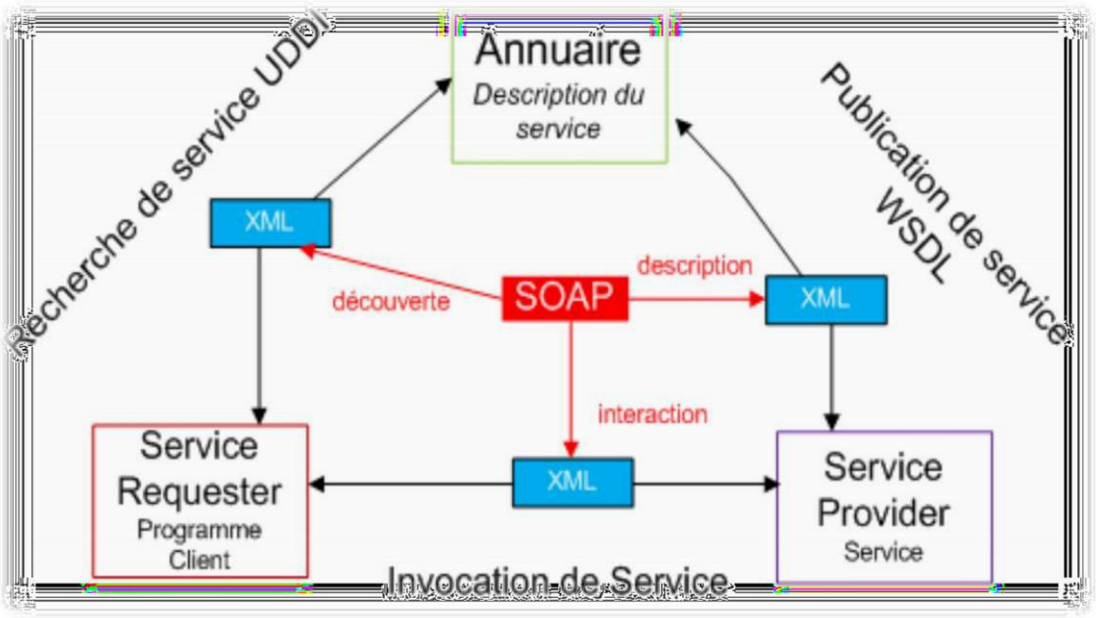


Figure I.7 : Structure de données de l'annuaire UDDI.

8. Fonctionnement des services web:

Le fonctionnement des services Web s'articule autour de trois acteurs principaux illustrés dans la (figure I.8) (http://openclassrooms.com /coures/les-services-web).



FigureI.8 : Fonctionnement d'un service web.

1. Service provider service : Le fournisseur de services est chargé de développer et de mettre en œuvre le service Web, puis de le rendre disponible sur Internet. Cela implique la création de l'infrastructure nécessaire, la gestion de la sécurité et des performances pour assurer l'accessibilité et le bon fonctionnement du service pour les utilisateurs.

2. **Service requester programme client :** Un client demandeur de service représente tout utilisateur ou programme qui consomme un service Web existant. Ce demandeur interagit avec le service en ouvrant une connexion réseau et en envoyant une demande au service Web. Cette demande est généralement formulée en XML, en utilisant des protocoles tels que XML-RPC ou SOAP pour la communication avec le service.

3. **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 :

✓ **La publication du service :** le fournisseur de services diffuse les descriptions de ses services Web dans l'annuaire. Cela permet aux clients demandeurs de service de découvrir et d'accéder aux services disponibles via l'annuaire de services.

✓ **La recherche du service :** Lorsqu'un client recherche un service particulier, il s'adresse à un annuaire qui lui fournit les descriptions et les URL des services demandés. Ces informations permettent au client de localiser et d'invoquer les services souhaités de manière appropriée.

✓ **L'invocation du service :** Une fois que le client a récupéré l'URL et la description du service auprès de l'annuaire, il les utilise pour invoquer le service auprès du fournisseur de services. Cela permet au client d'accéder aux fonctionnalités offertes par le service via une interaction réseau appropriée.

9. Les avantages et les inconvénients d'un service web :

9.1. Les avantages:

- Les services Web favorisent l'interopérabilité entre différents logiciels opérant sur une multitude de plateformes.
- Les services Web s'appuient sur l'utilisation de normes et de protocoles ouverts.

- Dans la mesure du possible, les protocoles et les formats de données utilisés par les services Web sont en format texte. Cela facilite la compréhension globale du fonctionnement des échanges.
- Grâce au protocole HTTP comme fondement, les services Web peuvent opérer à travers de nombreux pare-feu sans nécessiter de modifications importantes des règles de filtrage.
- Les outils de développement, qui reposent sur ces normes, permettent la création automatique de programmes utilisant les services Web déjà existants.

9.2. Les inconvénients:

- Actuellement, dans certains domaines, les normes de services Web sont Relativement récentes.
- Comparativement à d'autres approches de l'informatique distribuée telles que RMI, CORBA ou DCOM, les services Web présentent des performances relativement inférieures.
- En exploitant le protocole HTTP, les services Web peuvent contourner les mesures de sécurité établies à travers les pare-feu.

10. Conclusion

Ce chapitre a abordé diverses définitions des services web, y compris leur architecture et les normes de communication entre ces services. Le prochain chapitre se penchera sur le problème de la sélection, en commençant par un exemple illustratif de sa pertinence, puis en examinant un état de l'art regroupant les différentes approches proposées pour résoudre ce problème.

Chapitre II

Sélection des services Web

1. Introduction:

Ces dernières années, le domaine informatique a connu une évolution significative, marquée par une croissance constante tant en taille qu'en complexité des logiciels. Cette expansion est attribuable à une demande croissante et plus complexe en termes de fonctionnalités. En réponse à cette évolution, diverses approches ont été développées pour améliorer la productivité et l'efficacité des logiciels. Parmi ces approches, l'approche à services a gagné en popularité de manière significative.

L'émergence de cette approche à services ainsi que des fournisseurs de services a été notable. Cependant, dans ce contexte dynamique, le défi de sélection des meilleurs services se pose avec acuité. En effet, l'avènement des services Web et des services associés aux équipements a engendré une demande accrue en dynamisme, où les services peuvent apparaître et disparaître de manière imprévisible. Par conséquent, il est impératif de développer des méthodes avancées pour une sélection dynamique des meilleurs services, répondant ainsi à un besoin plus pressant que jamais.

2. Sélection des services web :

2.1 Définition:

La sélection de services web, selon (Naceur & Fegragui sid ahmed, 2016), consiste à choisir le service le plus approprié en fonction d'un besoin spécifique. Cette sélection repose principalement sur la description du service, qui se présente sous la forme d'une interface détaillant un ensemble d'opérations et de propriétés caractérisant le service. Chaque service est évalué en fonction d'un ensemble de critères de qualité définis, et chaque critère est associé à une série de règles permettant de calculer sa valeur pour un service donné. Ces critères,

accompagnés de leur définition, de leur granularité et des règles de calcul correspondantes, contribuent à évaluer les services de manière approfondie.

2.2 Exemple de motivation (Choix d'un voyage) :

Dans cette section, nous présentons un scénario illustrant et motivant notre approche pour la sélection de services. Imaginons que notre objectif soit de planifier un voyage de deux semaines dans un pays chaud, à proximité et à moindre coût. Pour cela, nous avons besoin de recourir à six services distincts (voir figure **II**.1) (Y & Bekaddour.H):

- Un service fournissant des informations géographiques.

- Un service fournissant des informations touristiques.

- Un service fournissant des informations météorologiques.

- Un service fournissant des informations sur les billets d'avion.

- Un service fournissant des informations sur les hôtels.

- ✓ Location de voiture .

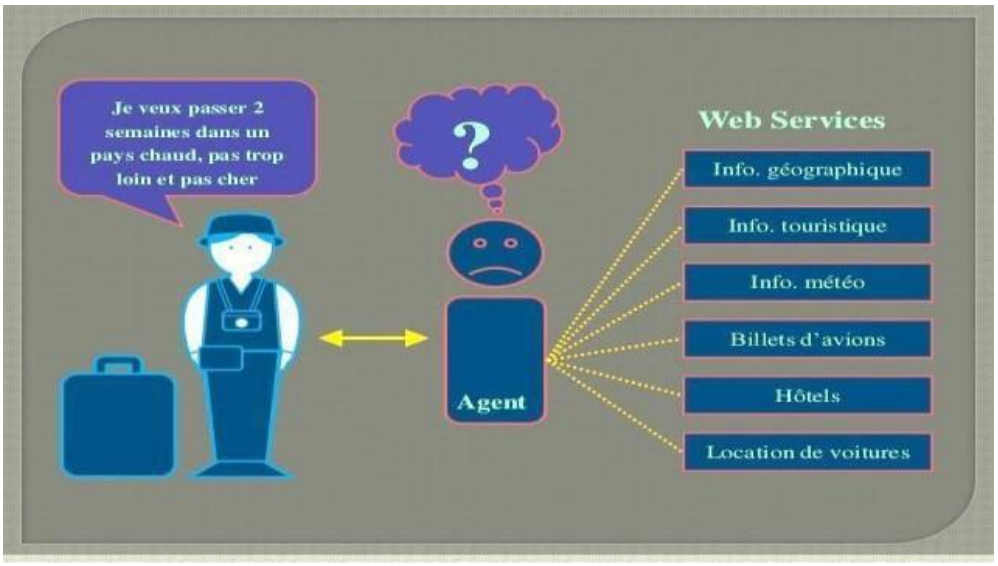


Figure II.1 : Exemple de motivation

Critères de QoS (Qualité de service) :

Cinq critères de qualité génériques sont pris en compte pour évaluer des services élémentaires :

- **Latence** : La latence mesure le retard prévu en secondes entre le début et la fin de l'exécution d'une opération. Elle est calculée par :

  Latence (Sol)= ∑n i=1 lat(Ci)

- **Coût** : Le coût d'exécution représente la somme d'argent que le demandeur de service doit payer pour l'exécution d'une opération. Les fournisseurs de service annoncent directement le prix d'exécution de leurs opérations. Le coût total est calculé par :

  Coût (sol) =∑n i=1 coût(Ci)

- **Réputation** : La réputation d'un service est une mesure de sa fiabilité, basée

principalement sur les expériences des utilisateurs. Les différents utilisateurs peuvent avoir des avis divers sur un même service. La valeur de la réputation est définie comme la moyenne des rangs attribués au service par les utilisateurs :

$$\mathbf{Rep\ (Sol) = 1/n \sum_{i=1}^{n} rep(Ci)}$$

- **Disponibilité** : La disponibilité d'un service représente la probabilité que le service soit accessible. Elle est calculée comme le logarithme en base 2 du produit des disponibilités individuelles des composants :

$$\mathbf{Disp\ (Sol) = log2 \prod_{i=1}^{n} disp\ (Ci)}$$

- **Fiabilité** : La fiabilité d'un service est la probabilité qu'une demande soit correctement traitée dans le délai maximum prévu. Elle est calculée comme le logarithme en base 2 du produit des fiabilités individuelles des composants :

$$\mathbf{Fiab\ (Sol) = log2 \prod_{i=1}^{n} fiab\ (Ci)}$$

On suppose que les domaines de valeurs des criteres de Qos sont défini comme suit

Critère	Domaine de valeurs
Cout	0 ..30 Euros
Temps d'execution	0..300 s
Réputation	0..5
Disponibilité	0 ..1
Fiabilité	0..1

Tableau II.1 : Les domaines de valeurs des critères

3. Optimisation combinatoire:

L'optimisation combinatoire est un outil essentiel qui combine différentes techniques de mathématiques discrètes et d'informatique pour résoudre des problèmes d'optimisation réels. Ces problèmes impliquent souvent la recherche de la meilleure solution dans un ensemble discret de solutions, appelé ensemble des solutions réalisables. Bien que cet ensemble soit généralement fini, sa taille peut être extrêmement grande.

Le but principal est de maximiser (problème de maximisation) ou de minimiser (problème de minimisation) une fonction objectif tout en respectant certaines contraintes. L'objectif est de trouver une solution optimale dans un laps de temps raisonnable (Halim).

Dans le contexte de l'architecture orientée services (SOA), la multitude de fournisseurs de services web, avec leurs différentes offres de services, rend difficile la tâche de sélection pour un concepteur cherchant à composer un ensemble de services web pour son système SOA. Étant donné que le nombre de combinaisons possibles est incalculable, le problème de sélection de services web devient difficile (NP-Difficile) et nécessite l'utilisation de solutions approximatives pour garantir une solution acceptable dans un temps raisonnable.

3.1 Optimisation combinatoire et SOA:

Dans le contexte de l'architecture orientée services (SOA), la multitude de fournisseurs de services web, avec leurs différentes offres de services, rend difficile la tâche de sélection pour un concepteur cherchant à composer un ensemble de services web pour son système SOA. Étant donné que le nombre de combinaisons possibles est incalculable, le problème de sélection de services web devient difficile (NP-Difficile) et nécessite l'utilisation de solutions approximatives pour garantir une solution acceptable dans un temps raisonnable.

Ainsi, le problème de sélection de services web peut être formulé comme un problème d'optimisation combinatoire en définissant une fonction objectif basée sur les critères de qualité de service requis par les architectes clients, et en caractérisant également les services web proposés par les différents fournisseurs.

3.2. Définition:

L'optimisation combinatoire vise à minimiser ou maximiser une fonction, souvent appelée fonction coût, de une ou plusieurs variables sous des contraintes spécifiques. Elle occupe une place cruciale dans la recherche opérationnelle, les mathématiques discrètes et l'informatique en raison de la complexité des problèmes d'optimisation et de leurs nombreuses applications pratiques. Bien que ces problèmes soient souvent facilement définis, ils sont généralement difficiles à résoudre, car la plupart appartiennent à la classe des problèmes NP-difficiles, pour lesquels il n'existe pas actuellement de solution algorithmique efficace pour toutes les instances (Halim).

Ainsi, un problème d'optimisation consiste à trouver le minimum ou le maximum d'une fonction donnée : $S^* = (f(s)/s \in S)$….. (Y & Bekaddour.H).

3.3. Approches d'optimisation combinatoire:

Pour l'optimisation de la sélection automatique de services, deux approches principales sont utilisées :

3.3.1 L'optimisation mono-objective:

Cette approche est privilégiée pour sa simplicité de mise en œuvre et sa grande généralité. Les objectifs du problème sont combinés en une seule fonction objective. Le décideur doit quantifier a priori l'importance de chaque critère pour construire cette fonction unique. Ensuite, le processus d'optimisation mono-objectif est lancé pour déterminer la solution "optimale". Un exemple courant de cette approche est la programmation linéaire

3.3.2 L'optimisation multi-objective:

Cette approche est née de la nécessité en industrie de satisfaire plusieurs critères contradictoires simultanément. Ses bases ont été posées par Pareto et Edgeworth au 19ème siècle, et elle trouve des applications en économie ainsi que dans les sciences de l'ingénieur. Contrairement à l'optimisation mono-objective, qui vise à optimiser une seule fonction objective, la plupart des problèmes réels sont caractérisés par plusieurs objectifs contradictoires à optimiser simultanément. L'optimisation multi-objective fournit alors un ensemble de solutions optimales, permettant aux décideurs de choisir le meilleur compromis pour résoudre le problème La figure suivante présente l'état de l'art des différents travaux menés sur le problème de sélection de services web.

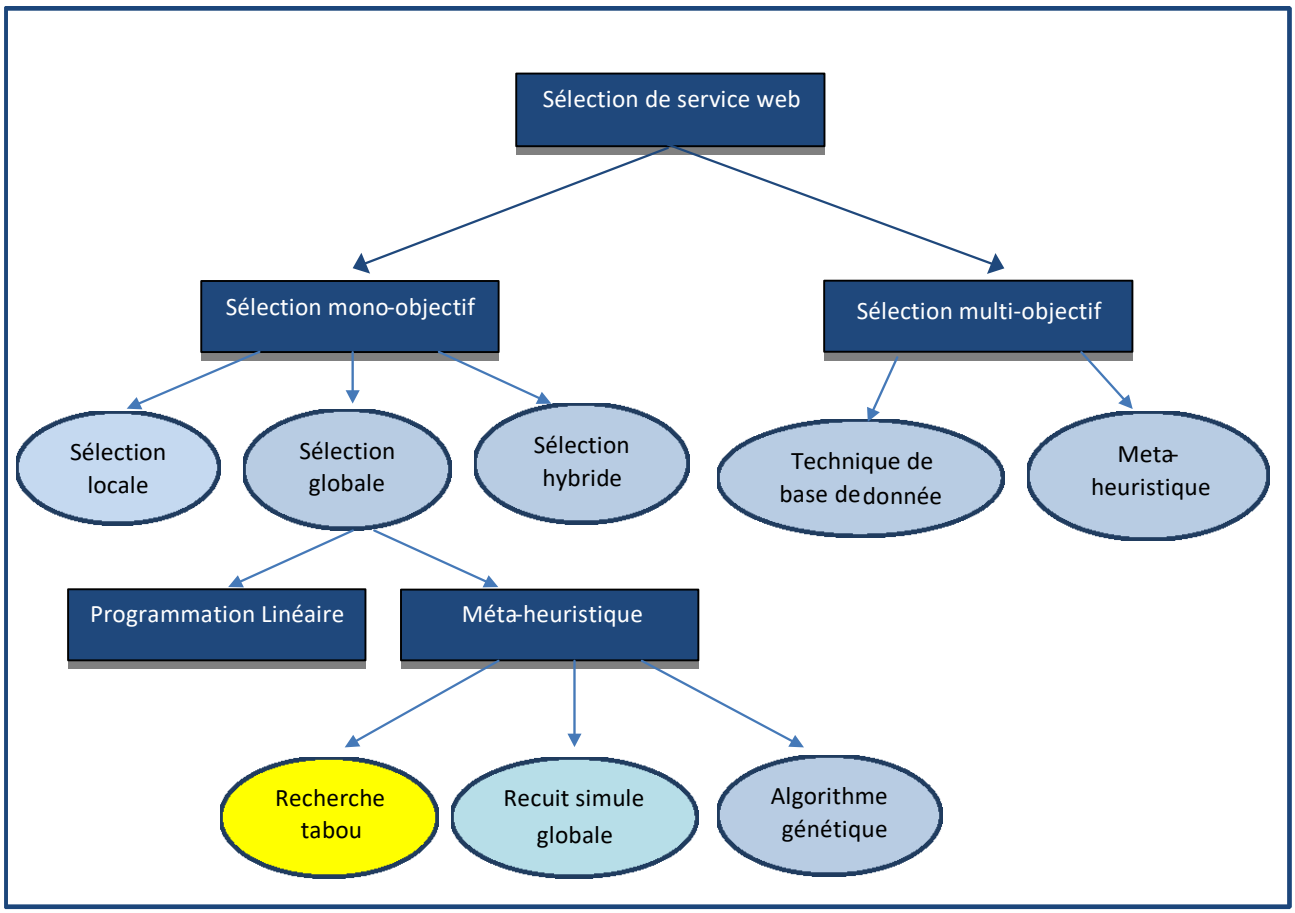


Figure II.2 : Les approches de Sélection

3.4. Les Méthodes d'optimisation combinatoire :

3.4.1 Méthodes de résolution:

Un large éventail de méthodes de résolution est disponible en recherche opérationnelle pour l'optimisation combinatoire. Nous pouvons les regrouper en deux grandes familles distinctes :

1. Les méthodes exactes garantissent la complétude de la résolution, c'est-à-dire qu'elles trouvent la solution optimale. Cependant, elles peuvent nécessiter des temps de calcul parfois prohibitifs, surtout pour les problèmes complexes.

2. Les méthodes approchées, quant à elles, peuvent ne pas garantir l'optimalité de la solution, mais elles fournissent souvent des solutions de qualité acceptable dans un laps de temps raisonnable. Ces méthodes sont particulièrement utiles pour les problèmes où la complexité rend difficile l'application de méthodes exactes.

Les méthodes de résolution peuvent être subdivisées comme suit (Maqrot):

- Méthodes exactes.
- Méthodes approchées.
- Heuristiques.
- Méta-heuristiques.
- Recherche locale.
- Méthodes évolutionnaires.

La Figure II.3 présente une classification visuelle de ces différentes méthodes d'optimisation.

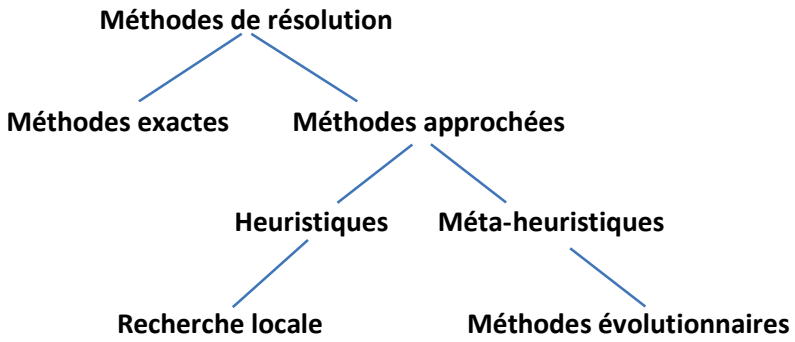


Figure II.3 – Classification des méthodes d'optimisation

3.4.1.1 Méthodes exactes:

Les algorithmes exacts sont utilisés pour trouver au moins une solution optimale d'un problème. Les algorithmes exacts les plus réussis dans la littérature appartiennent aux paradigmes de quatre grandes classes:

- La programmation dynamique,

- La programmation linéaire,

-Les méthodes de recherche arborescente (Branch&bound) (Halim).

3.4.1.1.1La programmation dynamique :

La programmation dynamique est basée sur le principe de Belmann "Si c'est un point qui appartient au chemin optimal entre A et B, alors la portion de ce même chemin allant de B à C'est le sous-chemin optimal entre B et C".

Cette approche consiste donc à construire les sous-chemins optimaux pour ensuite construire par récurrence le chemin optimal pour le problème entier. Dans la figure suivante, le chemin vert entre A et B est optimal. Ce chemin passe par le point C. Le sous-chemin vert [C, B] est donc optimal et le sous-chemin gris [C, B] ne peut exister car il est plus court que le sous-chemin

vert. Autrement dit, si le sous-chemin gris existe alors la solution optimale devient la séquence formée du sous-chemin vert [A, C] et du sous-chemin gris [C, B] au lieu du chemin vert [A, B].

La programmation dynamique s'appuie sur le principe de Bellman et consiste à prolonger le problème étudié en un problème plus général dont les paramètres sont entiers. Il faut généralement déterminer des relations de récurrence permettant de résoudre le problème d'un ordre donné en fonction de ses solutions pour les ordres inférieurs (Methosde exact en optimisation combinatoire)

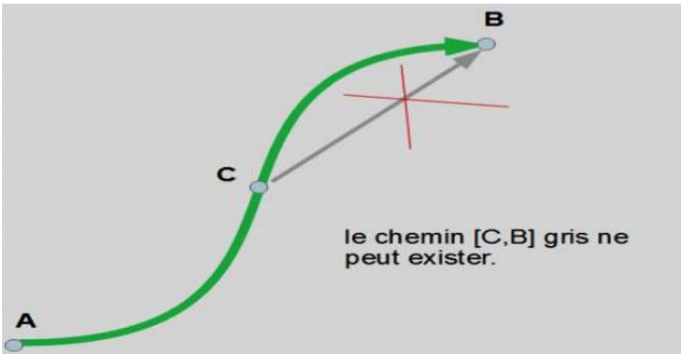


Figure II.4 Principe de Belman

3.4.1.1.2 La programmation linéaire :

Aborde la programmation linéaire (PL), un domaine d'optimisation visant à maximiser ou minimiser une fonction linéaire, appelée fonction objectif, sous contraintes linéaires exprimées par des équations ou des inéquations. Lorsque les variables doivent prendre des valeurs entières, on parle de problème de PL à variables entières (PLE) (Aissaoui & Souad ben hama).

3.4.1.1.3 Les méthodes de recherche arborescente (Branch&Bound):

La méthode Branch-and-Bound (procédure par évaluation et séparation progressive) qui consiste dans l'énumération des solutions possible d'une façon

intelligente. Cette technique arrive à éliminer des solutions partielles qui n'emmènent pas à la solution recherchée. L'algorithme proposé est divisé en deux phases:

- **Phase d'élagage:** qui consiste à filtrer les candidats qui ne sont pas assez bon pour chaque tâche.
- **Phase Branch-and-Bound:** Parcourt les combinaisons possibles
- Décomposition des candidats restant de la première phase, afin de trouver la meilleure composition des services (Naceur & Fegragui sid ahmed, 2016).

Les méthodes de résolution exactes possèdent quelques avantages et inconvénients, et le tableau suivant les représente (rima & kafi sara).

Les avantages	Les inconvénients
▯ Elles sont utilisées pour trouver au moins une solution optimale d'un problème donné.	▯ Utilisation pratique difficile (explosion combinatoire).
▯ Elles procèdent par énumération de toutes les solutions possibles afin de trouver la meilleure solution.	▯ Résolution des problèmes bicritères de petites tailles.

Tableau II.2 : Avantages et inconvénients des méthodes exactes.

3.4.1.2 Méthodes approchées (incomplètes) :

Lorsque les méthodes exactes échouent à résoudre les problèmes d'optimisation de grande taille, les méthodes approchées représentent une alternative efficace. Elles offrent la possibilité d'obtenir des solutions de qualité

satisfaisante dans un temps de calcul considérablement réduit. Ces approches reposent principalement sur des heuristiques, souvent spécifiques à un type de problème. Parmi ces méthodes, on trouve la catégorie des méta-heuristiques, qui se subdivise en deux grandes familles :

1) Les méthodes de recherche locale à solution unique, telles que la méthode de descente, le recuit simulé et la recherche tabou.

2) Les méthodes évolutionnaires basées sur une population de solutions, comme les algorithmes génétiques (Maqrot).

Les approches heuristiques:

Une heuristique est un algorithme qui ne donne pas toujours de solution.

Les méthodes heuristiques sont des approches de résolution de problèmes qui ne nécessitent pas une analyse exhaustive ou détaillée du problème. Elles fonctionnent en adoptant des approches successives, en se basant par exemple sur des similitudes avec des problèmes précédemment traités. Cette méthode permet d'éliminer progressivement les alternatives pour ne conserver qu'une série limitée de solutions, dans le but de converger vers une solution optimale. Les méthodes heuristiques sont souvent bien plus rapides que les méthodes exactes pour trouver une solution optimale.

3.5.Méta-heuristiques:

Les approches méta-heuristiques sont généralement des algorithmes stochastiques itératifs. Leur objectif est de converger vers un optimum global, c'est-à-dire l'extremum global d'une fonction, en échantillonnant cette fonction objectif. Ces méthodes agissent comme des algorithmes de recherche, cherchant à comprendre les caractéristiques d'un problème pour trouver une approximation de sa meilleure solution, d'une manière similaire aux algorithmes d'approximation.

Les méta-heuristiques se divisent en deux catégories:

3.5.1. Classification des Méta-heuristiques:

La résolution d'un problème d'optimisation s'effectue au moyen de méthodes d'optimisation, dont la classification est présentée dans la figure [**II**.4]. On distingue d'abord l'optimisation continue de l'optimisation discrète (ou combinatoire). Cette première distinction se réfère à la nature des variables utilisées dans le problème.

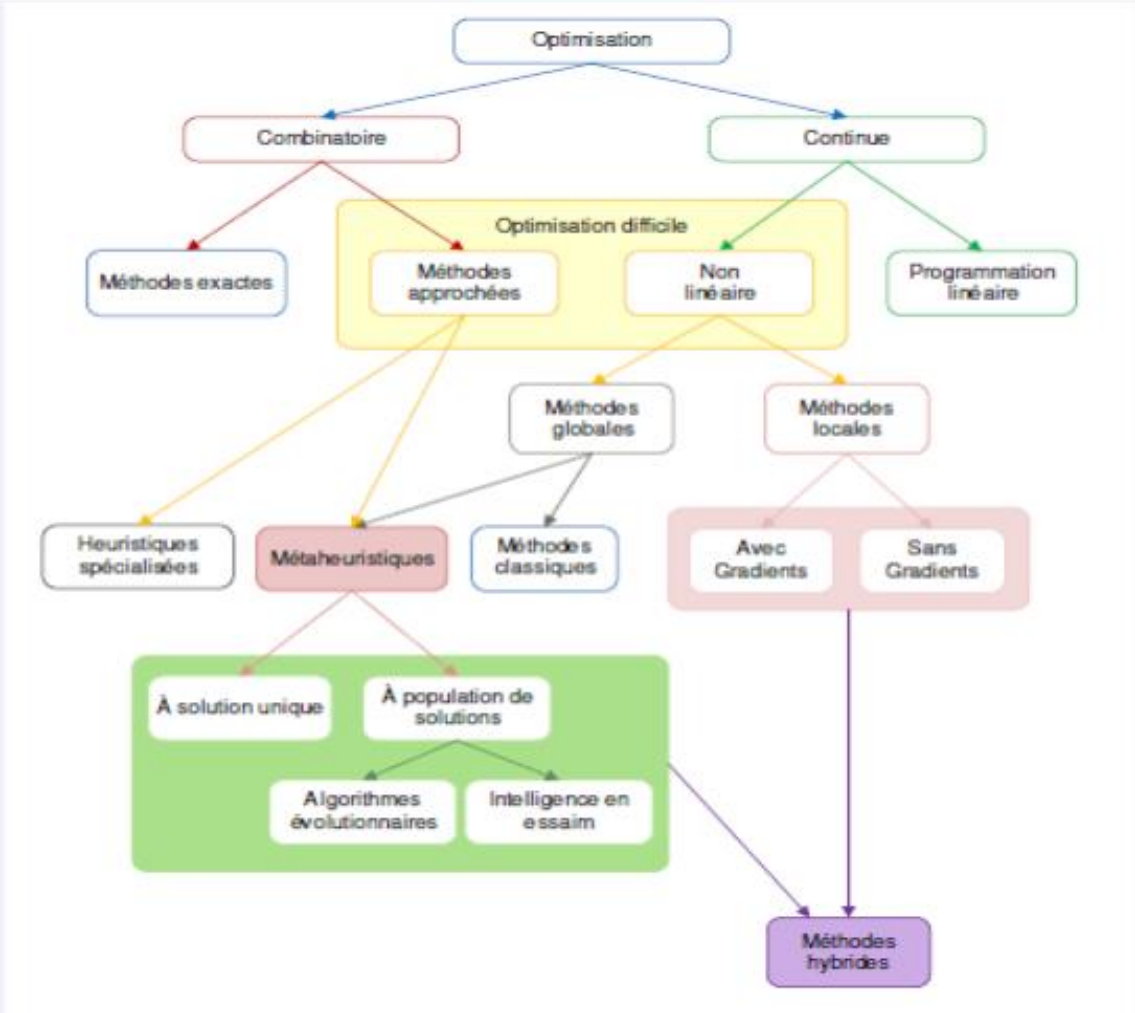


Figure II.5: Classification des méthodes d'optimisation [Dréo et al., 2003]

3.5.1.1 Les métas heuristiques à solution unique:

Les méta heuristiques à solution unique, également connues sous le nom de méthodes de trajectoire, débutent avec une seule solution initiale et évoluent progressivement en construisant une trajectoire dans l'espace de recherche. Cette catégorie englobe diverses méthodes, notamment la méthode de descente, la méthode du recuit simulé, la recherche tabou, la méthode GRASP, la recherche à voisinage variable, la recherche locale itérée, et leurs variantes.

3.5.1.1.1 Méthode de descente:

La méthode de descente, ou hill climbing dans les problèmes de maximisation, est l'une des méthodes les plus simples de la littérature. Elle part d'une solution initiale et choisit à chaque étape un point dans le voisinage de la solution actuelle qui améliore strictement la fonction objectif. Cependant, cette méthode peut rester piégée dans le premier optimum local rencontré, sans aucune forme de diversification. Une amélioration consiste à redémarrer plusieurs fois à partir d'une nouvelle solution générée aléatoirement lorsqu'un optimum local est trouvé.

3.5.1.1.2 Le recuit simulé (Simulated Annealing):

Le recuit simulé**,** quant à lui, trouve ses origines dans le formalisme de la mécanique statistique. Inspirée du processus de recuit physique utilisé en métallurgie, cette méthode vise à minimiser une fonction objective en introduisant une température fictive contrôlée par un schéma de refroidissement. Les mécanismes d'intensification et de diversification sont contrôlés par cette température, qui diminue progressivement au cours du processus de recherche. Les méthodes d'acceptation avec seuil sont des variantes du recuit simulé, où la décision d'accepter une dégradation est prise de manière déterministe en se basant sur une fonction auxiliaire et un seuil.

3.5.1.2.3 La méthode de recherche avec tabous:

La méthode de recherche tabou utilise explicitement l'historique de la recherche pour échapper aux minima locaux et mettre en œuvre une stratégie d'exploration. Elle utilise une mémoire appelée liste tabou pour enregistrer les dernières solutions rencontrées vers lesquelles il est interdit de se déplacer. Cette méthode peut être améliorée en incorporant un mécanisme d'aspiration et en utilisant des structures de mémoire à moyen et long terme pour approfondir les notions d'intensification et de diversification. (Maqrot).

3.5.1.2. Les méta-heuristiques à base de population de solutions :

Les méta heuristiques à population de solutions diffèrent des méthodes basées sur une solution unique en ce sens qu'elles améliorent progressivement une population de solutions au fil des itérations. Cette catégorie comprend les algorithmes évolutionnaires, inspirés de la théorie de l'évolution de Charles Darwin, et les algorithmes d'intelligence en essaim, qui tirent également leur inspiration de phénomènes biologiques naturels

3.5.1.2.1 Les algorithmes évolutionnaires:

Les algorithmes évolutionnaires (EA : Evolutionary Algorithms) s'inspirent de la théorie de l'évolution "darwinienne" pour résoudre une variété de problèmes. Selon Darwin, l'évolution des espèces résulte de la sélection naturelle, qui favorise les individus les mieux adaptés à leur environnement, ainsi que des variations génétiques aléatoires. Les EA englobent plusieurs métas heuristiques tels que les algorithmes génétiques, les stratégies d'évolution, la programmation évolutive et la programmation génétique. Le processus évolutionnaire des EA implique la génération d'une population d'individus, leur reproduction, la variation génétique, la sélection naturelle et le remplacement, répété sur plusieurs générations jusqu'à l'accomplissement d'un critère d'arrêt.

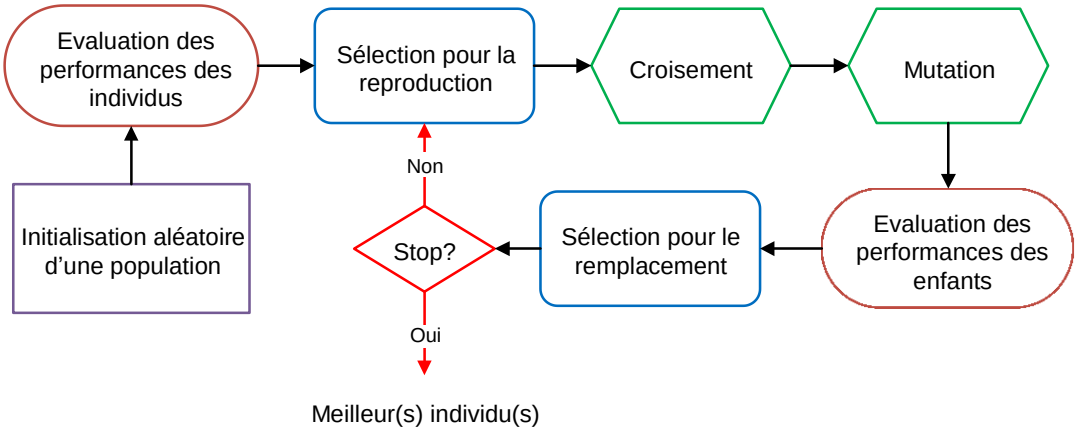


Figure* II.6*: Principe d'un algorithme évolutionnaire (EA) [Dréo et al., 2003]

3.5.1.2 .1.1 les algorithmes génétiques :

Les algorithmes génétiques (GA : Genetic Algorithms) sont parmi les méta heuristiques les plus populaires et largement utilisées des EA. Leurs origines remontent aux travaux de John Holland dans les années 1970. Les GA se distinguent par la représentation des données génétiques, initialement sous forme de vecteurs binaires ou de chaînes de caractères. Chaque étape des GA comprend des opérateurs tels que la sélection, le croisement, la mutation et le remplacement, qui guident l'évolution de la population vers de meilleures solutions.

3.5.1.2 .1.2La Stratégie d'Évolution

La Stratégie d'Évolution (ES : Evolution Strategy) a été introduite par Rechenberg dans les années 1960 et développée par Schwefel. Elle propose différentes variantes, notamment le ((1 + 1)-ES)-ES et le (μ, λ)-ES, qui impliquent plusieurs parents dans la génération d'un enfant. L'ES peut également adapter la taille de sa population en fonction de la performance des individus.

La programmation évolutionnaire (EP : Evolutionary Programming) a été introduite par L. J. Fogel dans les années 1960 comme une approche de l'intelligence artificielle. Elle se distingue par l'utilisation exclusive des opérateurs de mutation et de remplacement, sans recours au croisement.

La programmation génétique (GP : Genetic Programming) est une méthode automatisée pour créer des programmes informatiques à partir d'une spécification de haut niveau du problème. Elle utilise une représentation par arbres et des opérateurs tels que la mutation, le croisement et la reproduction pour évoluer vers des solutions meilleures.

L'intelligence en essaim L'intelligence en essaim (SI : Swarm Intelligence) est née de la modélisation mathématique et informatique des phénomènes biologiques rencontrés en éthologie [Bonabeau et al., 1999]. Elle recouvre un ensemble d'algorithmes, à base de population d'agents simples (entités capables d'exécuter certaines opérations), qui interagissent localement les uns avec les autres et avec leur environnement. Ces entités, dont la capacité individuelle est très limitée, peuvent conjointement effectuer de nombreuses tâches complexes nécessaires à leur survie. Bien qu'il n'y ait pas de structure de contrôle centralisée qui dicte la façon dont les agents individuels devraient se comporter, les interactions locales entre les agents conduisent souvent à l'émergence d'un comportement collectif global et auto-organisée.

Deux exemples phares d'algorithmes de l'intelligence en essaim sont les algorithmes de colonies de fourmis et les algorithmes d'optimisation par essaim particulaire.

3.5.1. 1 .3 L'optimisation par colonie de fourmis (ACO) :

L'optimisation par colonie de fourmis (ACO) est une technique d'optimisation inspirée du comportement des fourmis cherchant de la nourriture. Dans cette méthode, un ensemble de fourmis artificielles est utilisé pour trouver la meilleure solution à un problème donné. Chaque fourmi suit un chemin de manière probabiliste, en utilisant des informations locales et des signaux de phéromones laissés par les autres fourmis (Algorithme de colonie de fourmis wikipidia, Fr,wikipidia.org).

Voici les principes de base de l'ACO :

- **Construction de solutions probabilistes** : Chaque fourmi construit une solution en sélectionnant itérativement les étapes suivantes en fonction de probabilités spécifiques. Ces probabilités sont influencées par des informations locales sur les solutions déjà construites et des signaux de phéromones.
- **Évaporation des phéromones:** Les phéromones laissées par les fourmis sur les chemins diminuent progressivement avec le temps. Cela permet d'éviter que les fourmis suivent toujours les mêmes chemins.
- **Renforcement des chemins :** Les chemins qui mènent à de bonnes solutions sont renforcés par l'ajout de phéromones. Ainsi, les chemins les plus courts ou les plus efficaces ont tendance à être choisis par plus de fourmis, renforçant ainsi leur attractivité.
- **Exploration et exploitation :** L'ACO combine à la fois l'exploration de nouveaux chemins et l'exploitation des chemins déjà connus pour trouver la meilleure solution possible.

L'ACO a été largement appliquée à divers problèmes d'optimisation, tels que le voyageur de commerce, la conception de réseaux de télécommunications,

la planification de trajets logistiques, etc. Elle est appréciée pour sa capacité à trouver des solutions de haute qualité dans des espaces de recherche complexes.

3.5.1. 1 .2 L'optimisation par essaims particulaire PSO :

Cette technique est souvent décrite comme une sorte d'algorithme évolutionnaire, avec une population d'individus (les agents), dans laquelle, à chaque pas de temps, les « meilleurs » (selon un critère prédéfini) sont plus ou moins imités par les autres. Un aspect essentiel, qui la différencie, par exemple, des algorithmes génétiques classiques, est l'existence d'une mémoire, à laquelle ne contribuent que les meilleurs éléments. Le modèle est censé expliquer et reproduire certains comportements sociaux, mais s'est révélé un peu simpliste.

3.5.1. 2 Les méta heuristiques à solution unique :

Les méta heuristiques à solution unique, également connues sous le nom de méthodes de trajectoire, débutent avec une seule solution initiale et évoluent progressivement en construisant une trajectoire dans l'espace de recherche. Cette catégorie englobe diverses méthodes, notamment la méthode de descente, la méthode du recuit simulé, la recherche tabou, la méthode GRASP, la recherche à voisinage variable, la recherche locale itérée, et leurs variantes (benaichouch).

3.5.1.2.1 Méthode de descente (Hill Climbing):

La méthode de descente, ou hill climbing dans les problèmes de maximisation, est l'une des méthodes les plus simples de la littérature. Elle part d'une solution initiale et choisit à chaque étape un point dans le voisinage de la solution actuelle qui améliore strictement la fonction objectif. Cependant, cette méthode peut rester piégée dans le premier optimum local rencontré, sans aucune forme de diversification. Une amélioration consiste à redémarrer plusieurs fois à partir d'une nouvelle solution générée aléatoirement lorsqu'un optimum local est trouvé (Maqrot).

3.5.1.2.2 Le recuit simulé (Simulated Annealing) :

Le recuit simulé**,** quant à lui, trouve ses origines dans le formalisme de la mécanique statistique. Inspirée du processus de recuit physique utilisé en métallurgie, cette méthode vise à minimiser une fonction objectif en introduisant une température fictive contrôlée par un schéma de refroidissement. Les mécanismes d'intensification et de diversification sont contrôlés par cette température, qui diminue progressivement au cours du processus de recherche. Les méthodes d'acceptation avec seuil sont des variantes du recuit simulé, où la décision d'accepter une dégradation est prise de manière déterministe en se basant sur une fonction auxiliaire et un seuil (Recuit simulé : Définition, Exemple Fonctionnement, site :24 pm.com).

3.5.1.2.3 La méthode de recherche avec tabous

La méthode de recherche **tabou** utilise explicitement l'historique de la recherche pour échapper aux minima locaux et mettre en œuvre une stratégie d'exploration. Elle utilise une mémoire appelée liste tabou pour enregistrer les dernières solutions rencontrées vers lesquelles il est interdit de se déplacer. Cette méthode peut être améliorée en incorporant un mécanisme d'aspiration et en utilisant des structures de mémoire à moyen et long terme pour approfondir les notions d'intensification et de diversification (Maqrot).

4. Conclusion :

En conclusion, ce chapitre a mis en lumière le contexte scientifique dans lequel s'inscrivent les travaux de ce mémoire. De nombreux problèmes du monde réel peuvent être formulés comme des problèmes d'optimisation combinatoire, qu'il s'agisse de logistique, de production, d'extraction de connaissances, et bien d'autres encore. Cependant, résoudre ces problèmes d'optimisation est souvent ardu. Deux catégories de méthodes existent à cet effet : les méthodes exactes, garantissant la découverte de la meilleure solution réalisable mais généralement demandant un temps de calcul exponentiellement croissant avec la taille du problème, et les méthodes approchées, telles que les méta-heuristiques, qui visent à être plus rapides et à se rapprocher de la meilleure solution réalisable sans garantie absolue. Dans ce mémoire, ce sont ces dernières méthodes qui suscitent notre intérêt.

Chapitre III
Recherche Tabou

Chapitre III : Recherche Tabou

1. Introduction:

L'optimisation combinatoire est un domaine crucial en informatique et en recherche opérationnelle, nécessitant des techniques efficaces pour trouver les meilleures solutions parmi de nombreuses possibilités. La recherche Tabou, développée par Fred W. Glover en 1986, est une méthode d'optimisation méta heuristique particulièrement efficace dans ce contexte.

La recherche Tabou étend les techniques de recherche locale en utilisant une liste Tabou pour mémoriser les mouvements récents et éviter de revenir sur des solutions déjà explorées. Cela permet de surmonter les minima locaux et d'explorer l'espace de recherche plus efficacement.

Cette technique a été appliquée avec succès à divers problèmes complexes tels que le problème du voyageur de commerce (TSP), l'optimisation des réseaux, et la planification de la production. Avec l'essor des nouvelles technologies et des besoins en optimisation, la recherche Tabou reste une méthode pertinente et adaptable.

L'objectif de ce travail est de fournir une vue d'ensemble de la recherche Tabou, couvrant ses principes fondamentaux, ses étapes de mise en œuvre, ses avantages et limites, ainsi que ses applications pratiques. Nous aborderons également les développements récents et les extensions de l'algorithme.

2. Concepts de base

La recherche Tabou repose sur plusieurs concepts fondamentaux qui définissent son fonctionnement et son efficacité. Voici une explication détaillée de ces concepts :

2.1 Recherche Locale :

La recherche locale est une approche de résolution de problèmes qui consiste à explorer les solutions voisines d'une solution initiale pour trouver une solution optimale. Elle se concentre sur l'amélioration incrémentielle en évaluant et en modifiant les solutions existantes pour se rapprocher de l'optimum local. La recherche locale est au cœur de la méthode de recherche tabou, car elle permet d'explorer efficacement l'espace de recherche en se concentrant sur les solutions proches de la solution actuelle.

2.2 Liste Tabou :

La liste tabou est une mémoire adaptative utilisée dans la recherche tabou pour enregistrer les mouvements récents effectués dans l'espace de recherche. Elle sert à éviter de revisiter les solutions déjà explorées en interdisant certains mouvements pour un certain nombre d'itérations. Cela permet d'éviter les cycles et de diversifier l'exploration de l'espace de recherche. La liste tabou est mise à jour à chaque itération de l'algorithme en ajoutant les mouvements récents à la liste et en retirant les mouvements qui ont été autorisés pendant un certain nombre d'itérations.

2. 3 Critères d'Aspiration :

Les critères d'aspiration sont des conditions qui permettent de déroger aux règles de la liste tabou lorsque certaines conditions sont remplies. Par exemple, si une solution voisine améliore considérablement la qualité de la solution actuelle, elle peut être autorisée même si elle est normalement interdite par la liste tabou. Les critères d'aspiration permettent de garantir que l'algorithme ne reste pas bloqué sur des solutions sous-optimales et peut explorer de manière plus efficace l'espace de recherche.

2.4 .Diversification et Intensification :

La diversification et l'intensification sont deux mécanismes complémentaires utilisés dans la recherche tabou pour explorer efficacement l'espace de recherche. La diversification consiste à explorer de nouvelles zones de l'espace de recherche en générant des solutions radicalement différentes de la solution actuelle. Cela permet d'éviter de rester coincé dans des régions sous-optimales de l'espace de recherche. L'intensification, en revanche, consiste à approfondir l'exploration dans les zones prometteuses de l'espace de recherche en se concentrant sur les solutions les plus prometteuses. Ces deux mécanismes sont utilisés de manière équilibrée pour assurer une exploration complète et efficace de l'espace de recherche.

En comprenant ces concepts de base, on peut mieux appréhender le fonctionnement de la recherche tabou et son efficacité dans la résolution de problèmes d'optimisation combinatoire.

3. Étapes de l'algorithme

3.1 Pseudocode:

Initialiser la solution courante comme la meilleure solution trouvée jusqu'à présent

Initialiser la liste Tabou vide

Initialiser d'autres paramètres (nombre d'itérations maximum, critères d'arrêt, etc.)

Tant que le critère d'arrêt n'est pas atteint:

Générer les solutions voisines de la solution courante

Sélectionner la meilleure solution voisine non interdite par la liste Tabou

Mettre à jour la liste Tabou en ajoutant le mouvement effectué

Mettre à jour la solution courante avec la meilleure solution voisine

Vérifier les critères d'aspiration pour décider si un mouvement Tabou peut être accepté

3.1 Description détaillée :

****Initialisation** : Au début de l'algorithme, il faut initialiser la solution courante, souvent choisie de manière aléatoire, ainsi que la liste Tabou et d'autres paramètres comme le nombre maximum d'itérations ou les critères d'arrêt.

****Génération de voisins** : Pour chaque itération de l'algorithme, des solutions voisines sont générées à partir de la solution courante. Ces solutions

voisines sont des solutions légèrement modifiées par rapport à la solution courante, représentant ainsi une exploration locale de l'espace de recherche.

****Mise à jour**: Parmi les solutions voisines générées, la meilleure solution voisine qui n'est pas interdite par la liste Tabou est sélectionnée comme nouvelle solution courante. Cette étape vise à améliorer progressivement la qualité de la solution en explorant les mouvements autorisés par la liste Tabou.

****Mise à jour de la liste Tabou**: Après avoir sélectionné la meilleure solution voisine, le mouvement effectué est ajouté à la liste Tabou pour éviter de revisiter cette solution dans les itérations suivantes. Cela permet d'éviter les cycles et de diversifier l'exploration de l'espace de recherche.

4. Applications:

Domaines d'application:

La recherche Tabou trouve des applications dans une variété de domaines où des problèmes d'optimisation combinatoire se posent. Voici quelques-uns des domaines les plus courants où la recherche Tabou est appliquée:

1.**Problème du Voyageur de Commerce (TSP) :** La recherche Tabou est largement utilisée pour résoudre le problème du voyageur de commerce, qui consiste à trouver le chemin le plus court pour visiter un ensemble donné de villes une fois et revenir à la ville de départ. Les heuristiques de recherche Tabou sont efficaces pour trouver des solutions proches de l'optimum global dans des délais raisonnables.

2.**Planification et Ordonnancement :** La recherche Tabou est appliquée dans la planification et l'ordonnancement des tâches dans divers domaines tels que la production, la logistique, et les transports. Elle permet d'optimiser les

séquences d'activités pour maximiser l'efficacité opérationnelle et minimiser les coûts.

3**. Optimisation des Réseaux :** Dans les réseaux de télécommunications, les réseaux de distribution d'énergie, et d'autres domaines similaires, la recherche Tabou est utilisée pour résoudre des problèmes d'optimisation liés à la conception et à la gestion des réseaux. Elle peut être utilisée pour optimiser le routage des données, la distribution d'énergie, ou la conception de réseaux de transport.

- Études de cas:

Voici quelques exemples d'études de cas où la recherche Tabou a été appliquée avec succès :

1. Optimisation de la Chaîne Logistique : Une entreprise de logistique a utilisé la recherche Tabou pour optimiser la gestion de sa chaîne logistique, y compris la planification des itinéraires de livraison, l'affectation des véhicules, et la gestion des stocks. L'algorithme de recherche Tabou a permis de réduire les coûts de transport et d'améliorer l'efficacité opérationnelle.

2**. Optimisation des Horaires de Production :** Une usine de fabrication a utilisé la recherche Tabou pour optimiser les horaires de production afin de maximiser l'utilisation des équipements et minimiser les temps d'arrêt. L'algorithme a permis de réduire les temps d'attente et d'augmenter la production globale de l'usine.

3. Routage dans les Réseaux de Télécommunications : Une société de télécommunications a utilisé la recherche Tabou pour optimiser le routage des données dans son réseau, en tenant compte des contraintes de bande passante, de latence, et de redondance. L'algorithme a permis d'améliorer les performances du réseau tout en minimisant les coûts d'exploitation.

Ces études de cas illustrent la diversité des applications de la recherche Tabou et sa capacité à résoudre efficacement une gamme de problèmes d'optimisation dans différents domaines industriels.

****Critère d'aspiration:** Dans certains cas, un mouvement Tabou peut être autorisé malgré son interdiction par la liste Tabou si certaines conditions spécifiques, appelées critères d'aspiration, sont remplies. Cette étape permet de garantir que l'algorithme ne reste pas bloqué sur des solutions sous-optimales.

****Critère d'arrêt:** L'algorithme continue à itérer à travers les étapes précédentes jusqu'à ce qu'un critère d'arrêt soit atteint, tel que le nombre maximum d'itérations ou l'absence d'amélioration significative de la solution. Cela permet de déterminer quand arrêter l'exécution de l'algorithme.

En suivant ces étapes, l'algorithme de recherche tabou peut efficacement explorer l'espace de recherche pour trouver des solutions de haute qualité tout en évitant les pièges des minima locaux

5. Avantages et limites de la recherche Tabou:

La recherche Tabou présente plusieurs avantages significatifs qui en font une méthode d'optimisation populaire dans de nombreux domaines. Cependant, comme toute méthode, elle comporte également certaines limites à prendre en compte.

5.1 Avantages :

- **Capacité à échapper aux minima locaux** : La recherche Tabou est efficace pour échapper aux minima locaux, ce qui lui permet de trouver des solutions de meilleure qualité dans un espace de recherche complexe et non linéaire.

- **Flexibilité et adaptation** : Elle est flexible et peut être adaptée à une grande variété de problèmes d'optimisation, grâce à sa capacité à intégrer des critères d'aspiration et à être combinée avec d'autres techniques d'optimisation.

- **Facilité de mise en œuvre**: L'algorithme de recherche Tabou est relativement simple à mettre en œuvre par rapport à d'autres techniques d'optimisation plus complexes, ce qui le rend accessible même pour les problèmes complexes.

- **Performances robustes**: La recherche Tabou a démontré des performances robustes dans de nombreux domaines d'application, offrant souvent des solutions de haute qualité avec un temps de calcul raisonnable

5.2 Avantages Limites :

1. Sensibilité aux paramètres : La recherche Tabou peut être sensible aux paramètres choisis, tels que la taille de la liste Tabou ou le nombre maximum d'itérations. Le réglage fin de ces paramètres peut être nécessaire pour obtenir de bons résultats, ce qui peut nécessiter une expertise supplémentaire.

2. Complexité computationnelle : Pour les problèmes de grande taille ou hautement complexes, la recherche Tabou peut nécessiter des ressources computationnelles importantes, en particulier si la taille de l'espace de recherche est très grande.

3. Dépendance aux caractéristiques du problème : L'efficacité de la recherche Tabou peut dépendre des caractéristiques spécifiques du problème, telles que sa structure et sa topologie. Certains problèmes peuvent être plus adaptés à d'autres techniques d'optimisation.

4. Nécessité d'une initialisation appropriée : La qualité de la solution initiale peut avoir un impact significatif sur les performances de la recherche

Tabou. Une initialisation inappropriée peut entraîner une convergence prématurée vers des solutions sous-optimales.

6. Conclusion:

La recherche Tabou, développée par Fred W. Glover en 1986, est une méthode d'optimisation efficace pour résoudre les problèmes combinatoires en évitant les minima locaux. Ses concepts clés incluent la recherche locale, la liste Tabou, les critères d'aspiration, la diversification et l'intensification. Appliquée avec succès dans des domaines tels que le TSP et la planification, elle offre des avantages comme sa flexibilité et sa facilité de mise en œuvre, malgré des limites telles que sa sensibilité aux paramètres.

Chapitre IV
La Conception du système

Introduction:

Dans le contexte actuel de l'ère numérique, les services Web jouent un rôle crucial dans divers domaines, notamment celui de la réservation en ligne. Que ce soit pour réserver des vols, des hôtels ou des locations de voitures, les utilisateurs cherchent des solutions qui répondent à leurs besoins en termes de qualité de service (QoS). La qualité de service englobe divers paramètres tels que la latence, la disponibilité, le coût et la fiabilité des services Web.

Le problème de la sélection et de la composition de services Web consiste à identifier les meilleurs services disponibles qui respectent les contraintes et les préférences des utilisateurs. Cependant, trouver la meilleure combinaison de services peut être complexe en raison du grand nombre de services disponibles et des divers critères de QoS à prendre en compte. Pour résoudre ce problème, les méthodes heuristiques et méta heuristiques, telles que la recherche tabou, sont largement utilisées en raison de leur capacité à fournir des solutions optimales ou quasi-optimales dans un temps raisonnable.

2. Données de la sélection des services web:

- Description de la base:

Nous avons considéré dans ce qui suit dix types de services web. Chaque service est proposé par un certain nombre de compagnies (fournisseurs de services). Aussi, chaque service offert est caractérisé par un certain nombre de critères de qualité de service. Nous utilisons cinq paramètres pour évaluer la qualité des services : la latence, la fiabilité, la disponibilité, le coût et la réputation. Nous avons saisi manuellement toutes les valeurs afin d'assurer le maximum de combinaisons de ces critères. Un échantillon de notre base est illustré dans le tableau IV.1.

2.1. Expression de la fonction objective:

En considérant les critères cités ci-dessus, la fonction objective f(x) est définie comme une fonction des valeurs des attributs de la qualité de service QoS. Les valeurs de cette fonction correspondent aux différentes sélections qui peuvent être effectuées pour classer et composer la solution globale, en tenant compte des contraintes globales et des besoins des utilisateurs. A titre d'exemple et en considérant une solution x = S et les critères QoS, les contraintes peuvent être :

Coût (S) ≤ 2500 dinars : Le coût global d'une solution doit être inférieur ou égal à 2500 dinars.

Latence (S) ≤ 1000ms : La latence globale doit être inférieure ou égale à 1000ms.

Réputation (S) ≥ 50 % : La réputation globale doit être supérieure ou égale à 50 %.

Disponibilité (S) ≥ 50 % : La disponibilité globale doit être supérieure ou égale à 50 %.

Fiabilité (S) ≥ 50 % : La fiabilité globale doit être supérieure ou égale à 50 %.

Les valeurs des critères doivent optimiser une fonction globale qui fonctionne en maximisant les critères positifs (Réputation, Disponibilité et Fiabilité) et en minimisant les critères négatifs (Coût et Latence). Dans ce travail, et contrairement à d'autres travaux qui utilisaient d'autres fonctions objectives, nous définissons une nouvelle fonction qui exprime le score d'une solution, et permet ainsi de classer les différentes solutions avec attention, et de rechercher la solution.

La fonction objective f pourrait s'exprimer indifféremment comme :

$$(f) = \Sigma\Sigma(Cik)k \in Pos \Sigma Log\ (Cik)k \in NegNi = 1$$

$$Min(g) = \frac{1}{Max(f) = \sum_{i=1}^{N} \frac{\sum_{k \in Pos} Log(C_i^k)}{\sum_{k \in Neg} Log(C_i^k)}}$$

Où

$$Max(f) = \sum_{i=1}^{N} \frac{\sum_{k \in Pos} Log(C_i^k)}{\sum_{k \in Neg} Log(C_i^k)}$$

k∈Pos est l'ensemble des critères positifs (Réputation, Disponibilité, Fiabilité).

k∈Neg est l'ensemble des critères négatifs (Coût, Latence).

Pos, Neg sont respectivement les ensembles de critères positifs et négatifs.

D'après l'expression de f, elle est directement proportionnelle aux variables de l'ensemble Pos, et inversement proportionnelle aux variables de l'ensemble Neg. Nous avons introduit le logarithme des critères afin que les produits ne dépassent pas la taille de représentation.

Campanie	Service	Coût	Latence	Fiabilité	Disponibilité	Réputation
1	6	1000	400	70	50	50
1	3	950	300	90	70	70
1	2	2100	700	50	70	70
2	1	2400	400	60	80	80
2	5	900	900	70	90	90
2	7	2000	500	80	90	90
2	3	700	450	90	70	70
3	4	500	500	70	70	70
3	6	2100	950	90	50	50

3	**9**	**800**	**1000**	**60**	**80**	**80**
2	**9**	**700**	**600**	**50**	**70**	**90**
1	**5**	**600**	**300**	**60**	**60**	**70**
1	**3**	**850**	**250**	**80**	**70**	**90**
4	**6**	**600**	**400**	**50**	**90**	**80**
5	**4**	**1600**	**650**	**70**	**90**	**70**
5	**1**	**1000**	**700**	**80**	**60**	**50**

Tableau IV.1 : Un exemple de base de données.

3. Diagrammes de conceptions:

La conception d'un projet informatique quelconque est la phase essentielle, dans laquelle une étude préalable globale est menée. L'objectif principal de cette phase est d'analyser l'ensemble des besoins puis d'imaginer des contextes d'utilisation.

Pour arriver à une bonne conception il est nécessaire de choisir la bonne méthode de conception, pour cela nous allons présenter la conception de notre projet sous forme de diagrammes UML.

Dans notre conception nous avons utilisé 3 types de diagrammes :

 Le diagramme de cas d'utilisation qui permet de représenter les deux acteurs de notre système et les tâches qui leurs sont associées.

 Le diagramme de classes, qui exprime les classes et les associations entre elles dans notre système.

 Le diagramme de séquences pour représenter les échanges de messages entre les acteurs et le système.

3.1. La vue statique:

3.1.1. Diagramme de cas d'utilisation:

Un diagramme de cas d'utilisation permet de recueillir, d'analyser et d'organiser les besoins des utilisateurs, et de recenser les grandes fonctionnalités d'un système.

.

Acteur	Rôle
Client	▪ Introduire les critères de sélection de service ▪ Consulter le service optimal
Gérant	▪ Définition des critères de filtrage ▪ Chargement de la BDD

Tableau IV.2 : Identification des acteurs et leurs rôles.

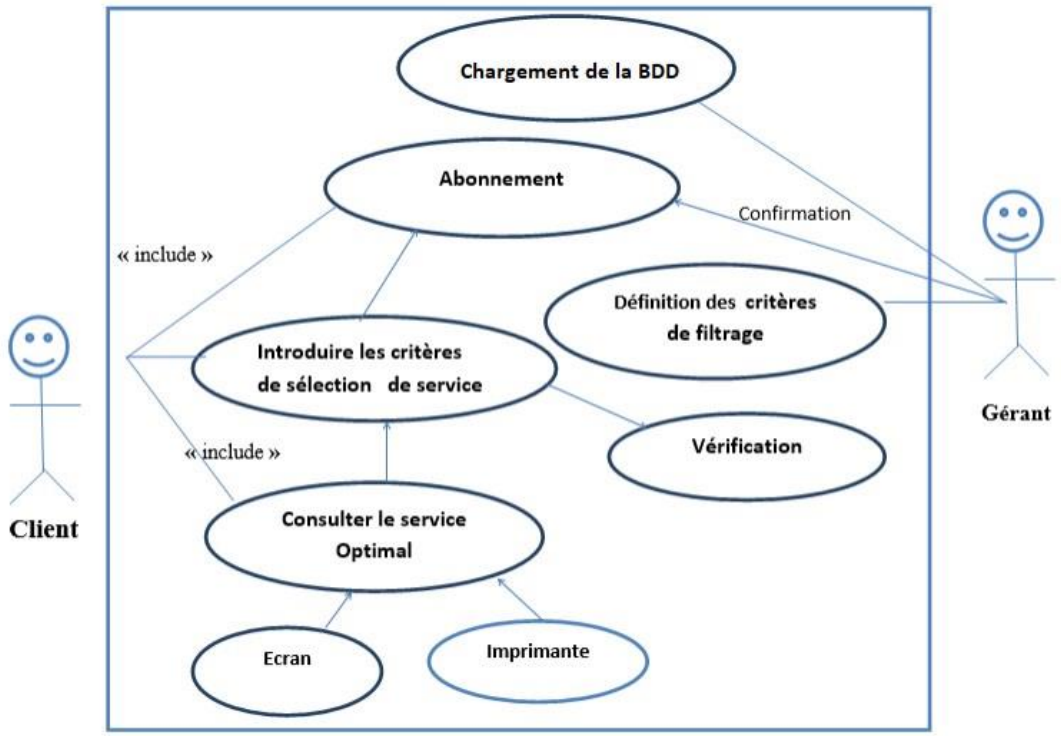


Figure IV.1 : Diagramme de cas d'utilisation.

3.1.2. Diagramme de classes:

Un diagramme de classes permet de représenter les classes intervenant dans le système, notamment celles en rapport avec les données qui sont utilisées. Il constitue un élément très important de la modélisation et permet de définir quelles seront les composantes du système final. Il permet de structurer le travail de développement de manière très efficace.

La figure IV.3 représente le diagramme de classes de notre application.

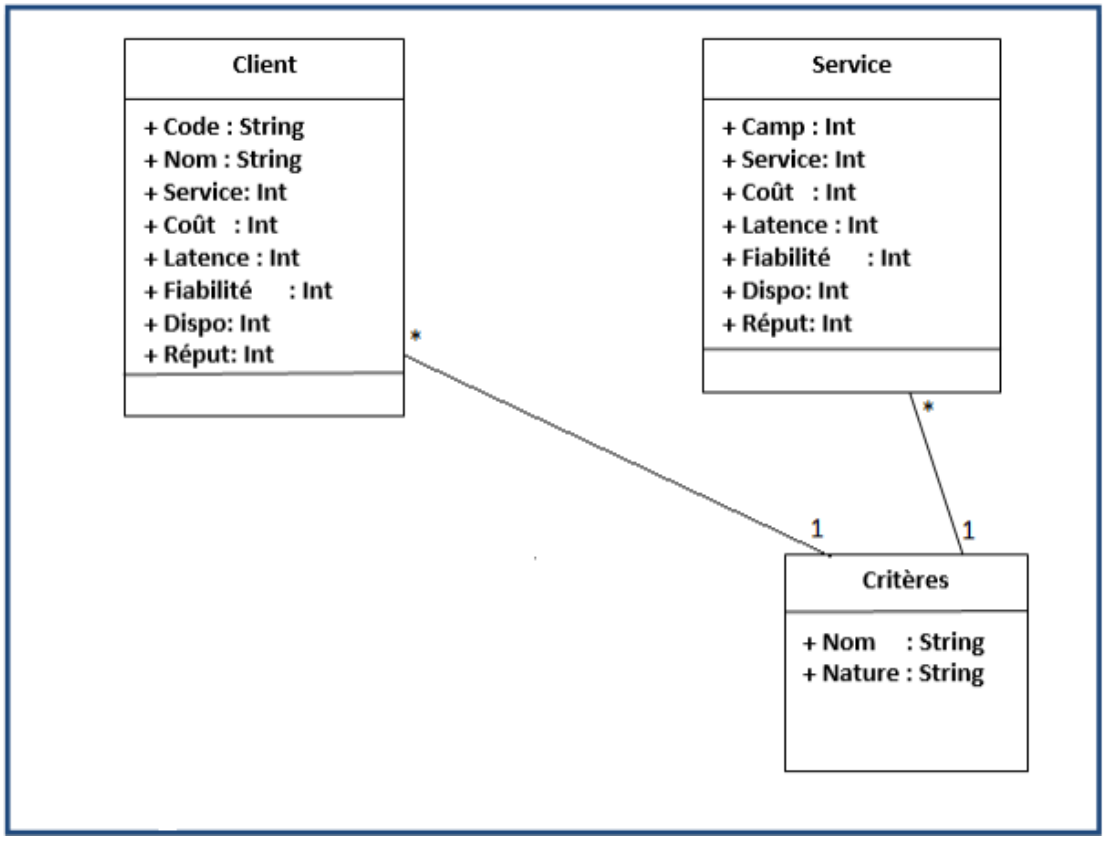


Figure IV.2 : Diagramme de classes.

➢ **Description des classes :**

▪ **La classe "Service" :** Regroupe tous les critères de qualité de service offerts par les différentes compagnies.

La classe "Client " : Regroupe tous les critères de qualité de service exigés par les Client.

La classe "Critère" : Permet de définir la nature des critères de qualité de service

1) Positive comme (La fiabilité, disponibilité, réputation).

2) Négative comme (Le coût, la latence).

3.2. La vue dynamique:

3.2.1.Diagramme de séquences:

C'est un diagramme dynamique, il permet de représenter la dynamique d'un use cas ou la collaboration d'un ensemble d'objets internes au système. Il montre la séquence, représentation verticale chronologique, des messages passés entre blocs au sein d'une interaction.

Description des diagrammes de séquences de chaque cas

Cas d'utilisation	Acteur principal	Pré condition
Scénario 1 : Introduire les critères de Sélection de service	Client	Le client doit être abonné. Le client atteint l'interface de saisie de données.
Scénario 2 : Définition de critères de filtrage	Gérant	Le gérant vérifie les critères de filtrage.
Scénario 3 : Chargement de la BDD	Gérant	Le gérant atteint l'interface de chargement de la BDD. Chargement de données client.
Scénario 4 : Consulter le service optimal	Client	Le client doit être abonné. Le client atteint l'interface de la meilleure solution.

Tableau IV.3 : Description de diagramme de séquences

Scénario 1 « Introduire les critères de sélection de service » : Ce cas d'utilisation commence lorsque le client veut introduire ses critères de sélection du service optimal.

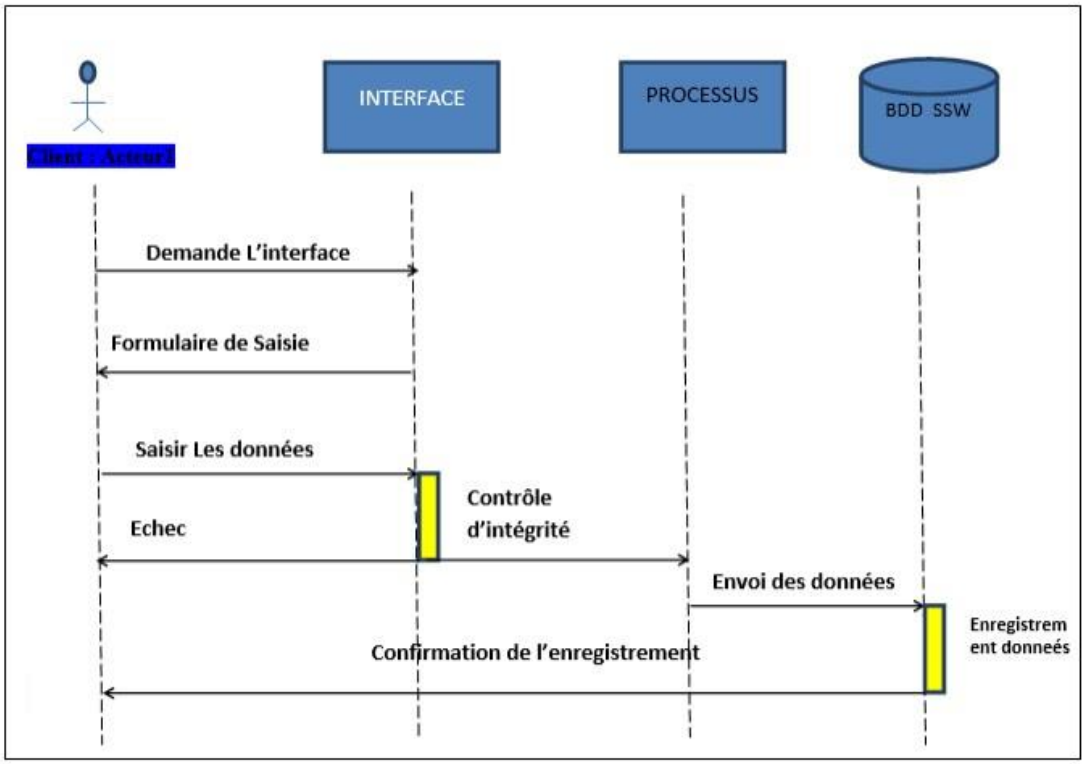


Figure IV.3 : Diagramme de séquences

« Introduire les critères de sélection de service».

Scénario 2 « Introduire les critères de filtrage » : Ce cas commence lorsque le gérant veut définir les critères de filtrage.

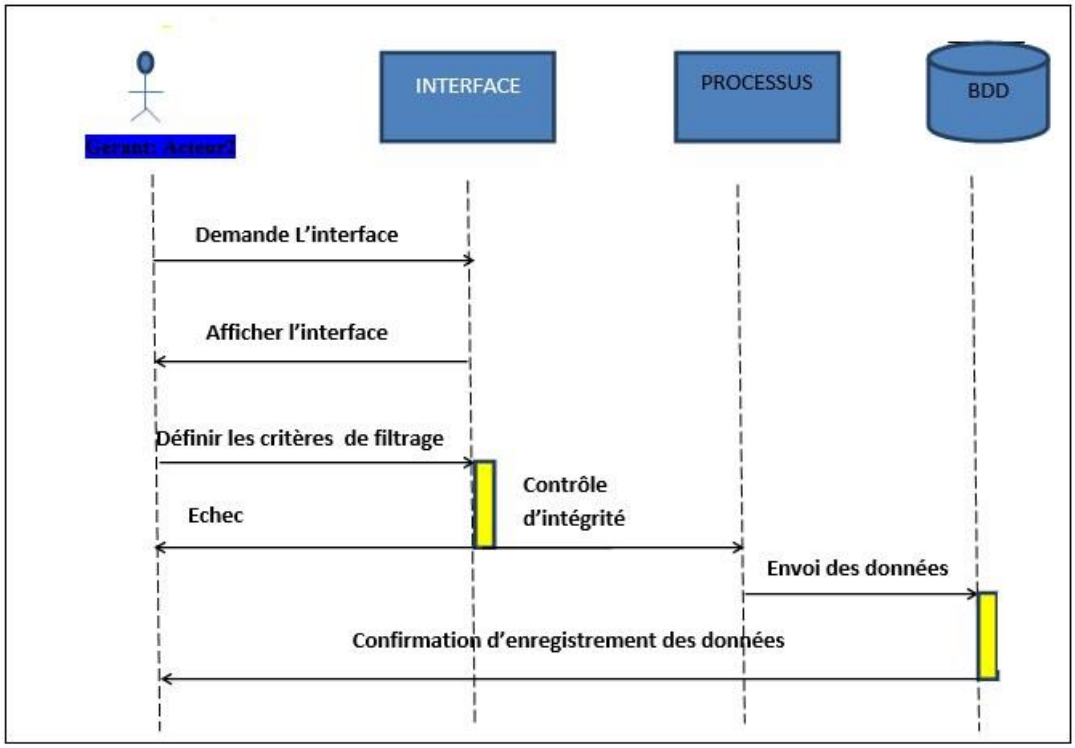


Figure IV.4 : Diagramme de séquences « Définition des critères de filtrage».

Scénario 3 « Chargement de la BDD » : Ce cas d'utilisation commence lorsque le gérant veut ajouter les données à la BDD.

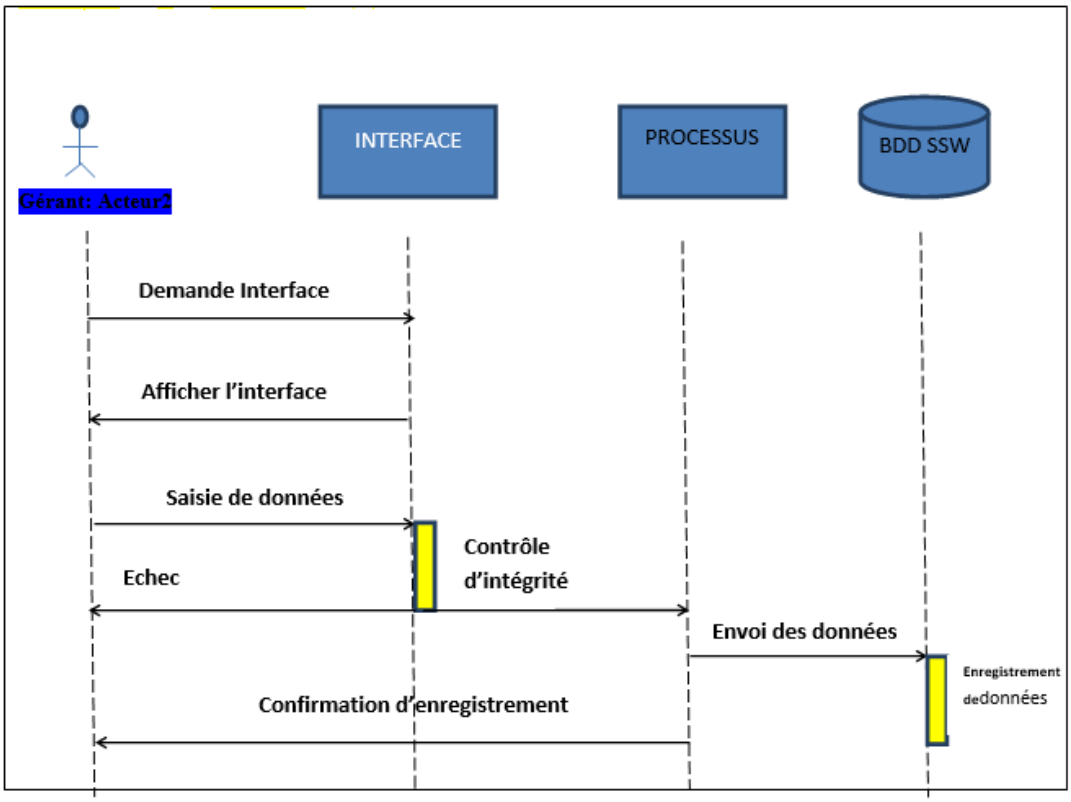


Figure IV.5 : Diagramme de séquences « Chargement de BDD».

Scénario 4 « Consulter le service optimal » : Ce cas d'utilisation commence lorsque le client veut consulter le service optimal selon les critères choisis.

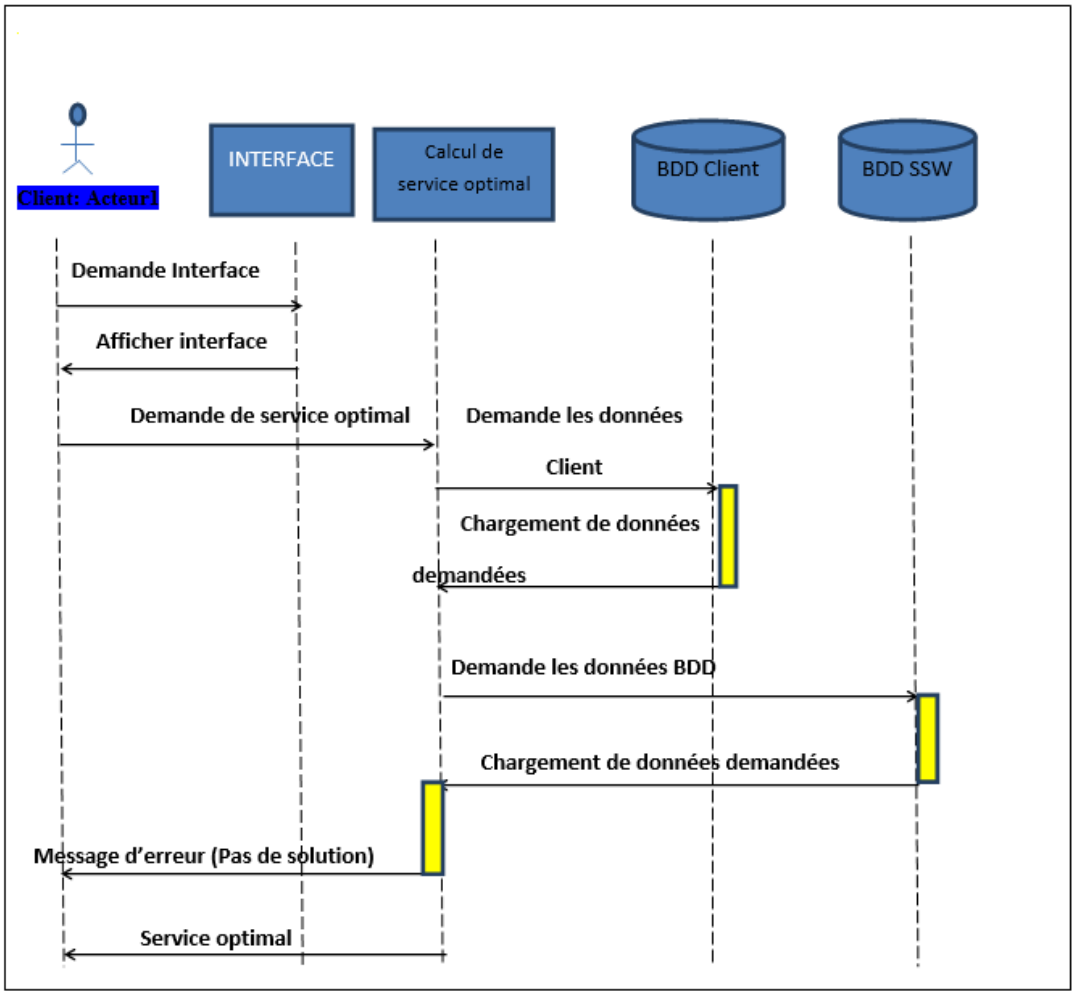


Figure IV.6 : Diagramme de séquences « Demande du calcul de la solution optimale ».

Conclusion:

Dans ce chapitre, nous avons présenté notre système en expliquant son fonctionnement et sa conception à l'aide du langage UML Les diagrammes issus de cette conception nous permettons de maitriser et comprendre toutes les vues de notre système, ce qui nous permettra par la suite de l'Implémenter facilement.

Chapitre V
Implémentation et Test

Chapitre V: Implémentation et Test

1. Introduction:

Ce chapitre représente l'ultime phase menant à l'achèvement de notre application. L'implémentation consiste à convertir les résultats de la phase de conception en un programme ou logiciel informatique fonctionnant sur une machine, en utilisant des outils de développement et de programmation appropriés pour résoudre le problème en question.

Nous allons détailler les éléments de l'environnement de travail, ainsi que les outils utilisés pour l'implémentation, et fournir quelques exemples d'interfaces développées dans notre logiciel.

2. Environnement de développement:

2.1. Environnement matériel:

- **PC portable DELL :**

2.2. Environnement logiciel (Langage de programmation):

Le langage de programmation utilisé pour le développement de notre application est le Delphi, une des versions du langage Pascal sous Windows.

2.2.1. Présentation du Delphi:

Créé en 1995 par Borland Software Corporation, Delphi est un langage de programmation orienté objet de haut niveau et un environnement de développement intégré (EDI) pour Windows. Une version pour Linux est sortie en 2001. Delphi propose des outils pour le développement de logiciels, comme un éditeur de texte, un compilateur et un débogueur. Depuis 2007, Delphi appartient à Embarcadero et est disponible pour Windows, Linux, macOS, iOS et Android (Delphi).

Delphi est particulièrement adapté pour :

- La programmation d'applications graphiques mobiles.
- La gestion de bases de données.
- Le développement de logiciels d'entreprise.

Figure V.1 : Logo de Delphi 7

2.2.2. Les avantage du Delphi:

- ***Rapidité de développement :*** Delphi permet une productivité accrue grâce à son environnement convivial, ses composants visuels et son langage de programmation intuitif, facilitant ainsi le développement rapide d'applications.

- ***Performance :*** Les applications créées avec Delphi sont réputées pour leur haute performance, surtout sur les plateformes Windows.

- ***Accès aux fonctionnalités Windows :*** Delphi permet d'exploiter pleinement les fonctionnalités du système d'exploitation Windows, offrant ainsi la possibilité de développer des applications robustes et intégrées à la plateforme.

- ***Héritage de Pascal :*** Le langage de programmation utilisé par Delphi, basé sur Pascal, est reconnu pour sa lisibilité et sa facilité d'apprentissage, ce qui en fait une option attrayante pour les débutants en programmation.

- ***Support de la programmation orientée objet :*** Delphi prend en charge la programmation orientée objet, ce qui facilite le développement d'applications modulaires, faciles à maintenir et à faire évoluer.

2.2.3. Les inconvénients du Delphi

- ***Plateforme spécifique :*** Le Delphi est principalement employé pour créer des logiciels Windows, ce qui restreint sa portabilité vers d'autres plateformes.

- ***Coût élevé :*** Les licences Delphi peuvent représenter un investissement significatif, ce qui peut constituer un obstacle pour les petites entreprises et les développeurs indépendants.

- ***Communauté restreinte :*** La communauté des développeurs Delphi est relativement plus restreinte par rapport à d'autres langages de programmation

populaires, ce qui peut rendre plus difficile la recherche de ressources et d'assistance en cas de besoin.

➢ ***Support pour les technologies émergentes :*** Delphi peut parfois prendre du retard dans le support des technologies et des frame works émergents les plus récents, ce qui peut restreindre les options de développement avancées.

2.2.4. Ce que distingue le Delphi des autres langages:

Le Delphi se distingue des autres langages de programmation par son étroite intégration avec la plateforme Windows et son environnement de développement convivial. Son accès direct aux fonctionnalités du système d'exploitation permet de concevoir des applications robustes. De plus, le langage de programmation Pascal, reconnu pour sa lisibilité, simplifie le développement de code clair. Grâce à son IDE complet, le Delphi favorise une productivité élevée, permettant aux développeurs de créer des applications rapidement et efficacement.

2.2.5. La version du Delphi:

La version du Delphi a été utilisée est Delphi 7.0

3. Présentation de l'interface de l'application:

Dans cette partie nous allons présenter le mode de fonctionnement et les différentes fenêtres de notre application. Commençant par la page d'accueil de l'interface, cette première fenêtre « **Présentation** » donne une idée générale sur l'objectif de notre application.

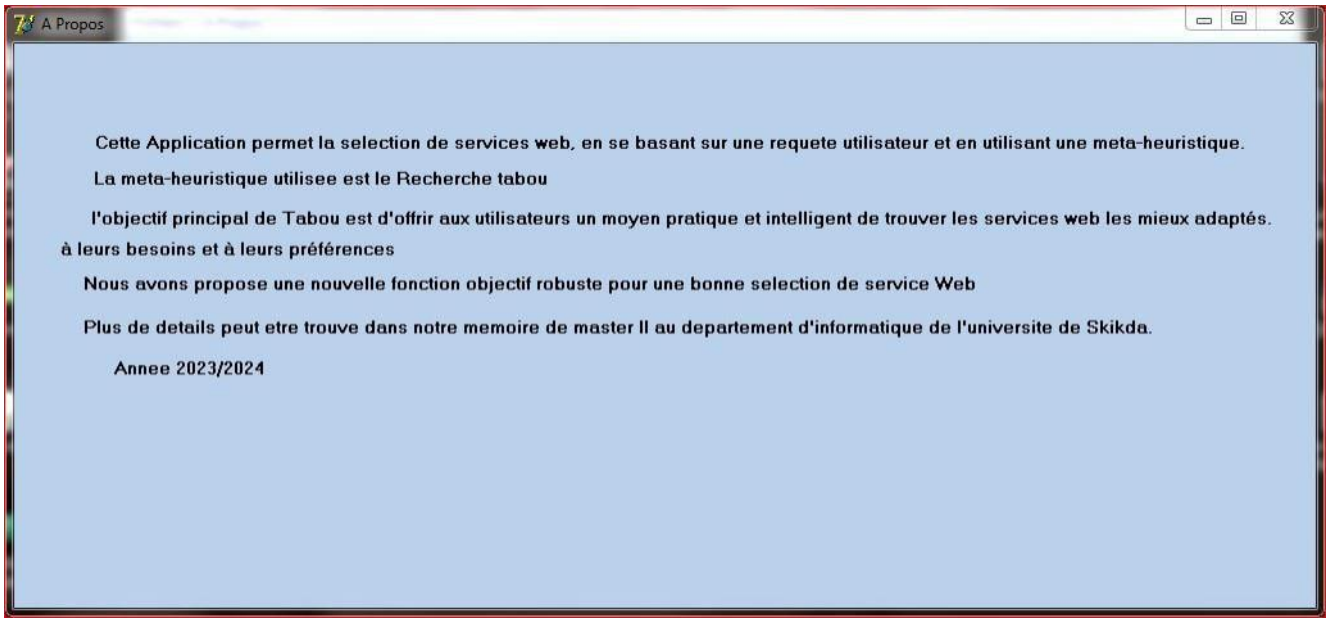


Figure V.2 : Page d'accueil.

La deuxième fenêtre présente la fenêtre principale de notre application. Elle est constituée de deux onglets Fichier et Présentation.

Figure V.3 : Fenêtre principale du système.

A partir du menu **Fichier** on peut accéder à quatre sous menu de notre application:

- **Données des services.**
- **Données du client.**
- **Services sélectionnés.**
- **Sortir.**

En cliquant sur le bouton « **Données des services »** le gérant peut accéder à la BDD de sélection de services web, où il peut charger et faire la mise à jour de la BDD (Figure V.4)

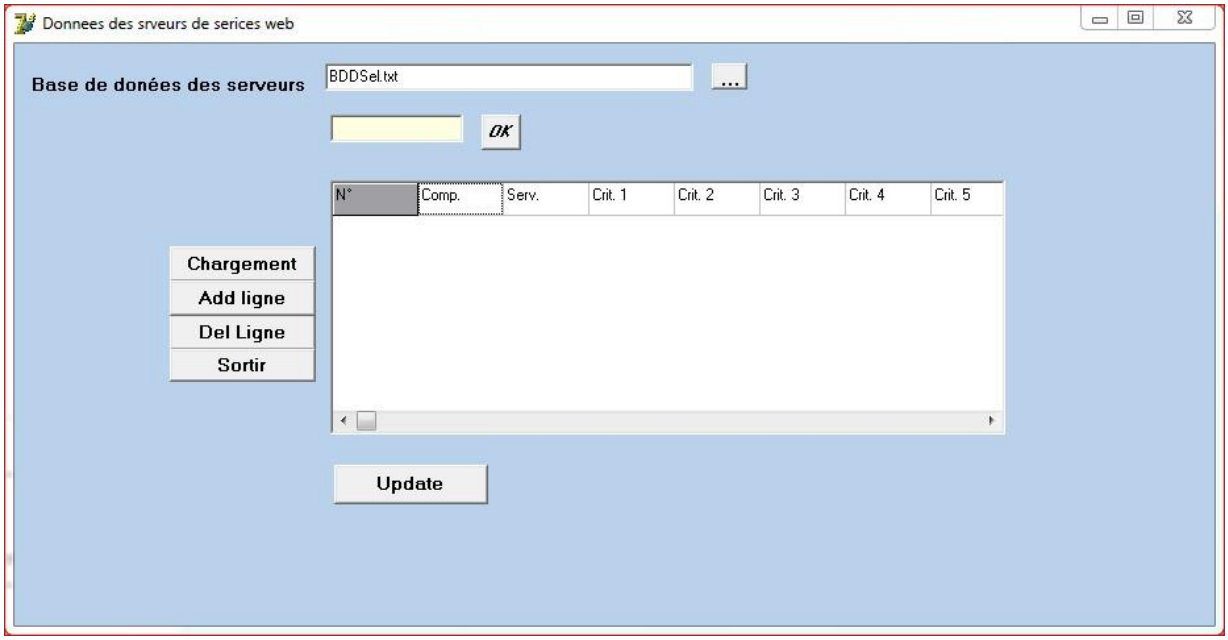


Figure V.4 : Interface Mise à jour BDD.

Pour accéder à l'espace client on doit cliquer sur le bouton « **Données du client »** Cet espace permet de sélectionner le fichier **C.txt** celui-ci est enregistré dans notre base de données **Client.** Le client va avoir une sélection optimale de services web en fonction de la BDD des services web. (Figure V.5).

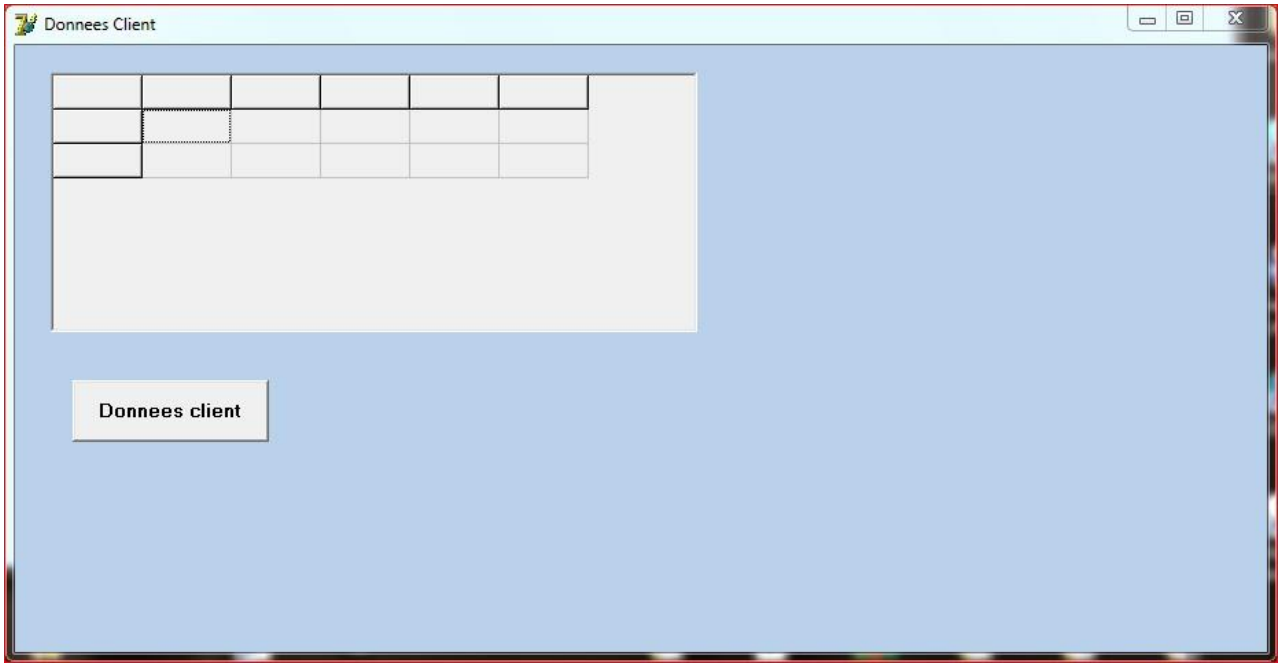


Figure V.5 : Chargement des données client.

Et pour calculer et afficher le service web optimal qui répond aux exigences de notre client avec notre Meta-heuristique « **Recherche Tabou »** on doit cliquer sur le bouton « **Services sélectionnés »** Ceci consiste à afficher le **numéro de la compagnie** et le **numéro du service** qui assure le service le plus proche aux critères introduits le client. (Figure V.6).

Figure V.6 : Interface Résultats de la sélection de services web.

Et à la fin et pour quitter notre application on doit cliquer sur le bouton « **Sortir** ».

Figure V.7 : Sortir de l'application.

4. Expérimentation:

Nous proposons une configuration qui se base sur la sélection de trois services, chaque service est caractérisé par les cinq contraintes de recherche tabou. Ces contraintes doivent être vérifiées durant la sélection des services web, en respectant ces dernières nous devons ne pas dépasser une certaine valeur de coût et latence, et nous devons garantir un maximum de disponibilité, fiabilité et réputation de ces services sélectionnés.

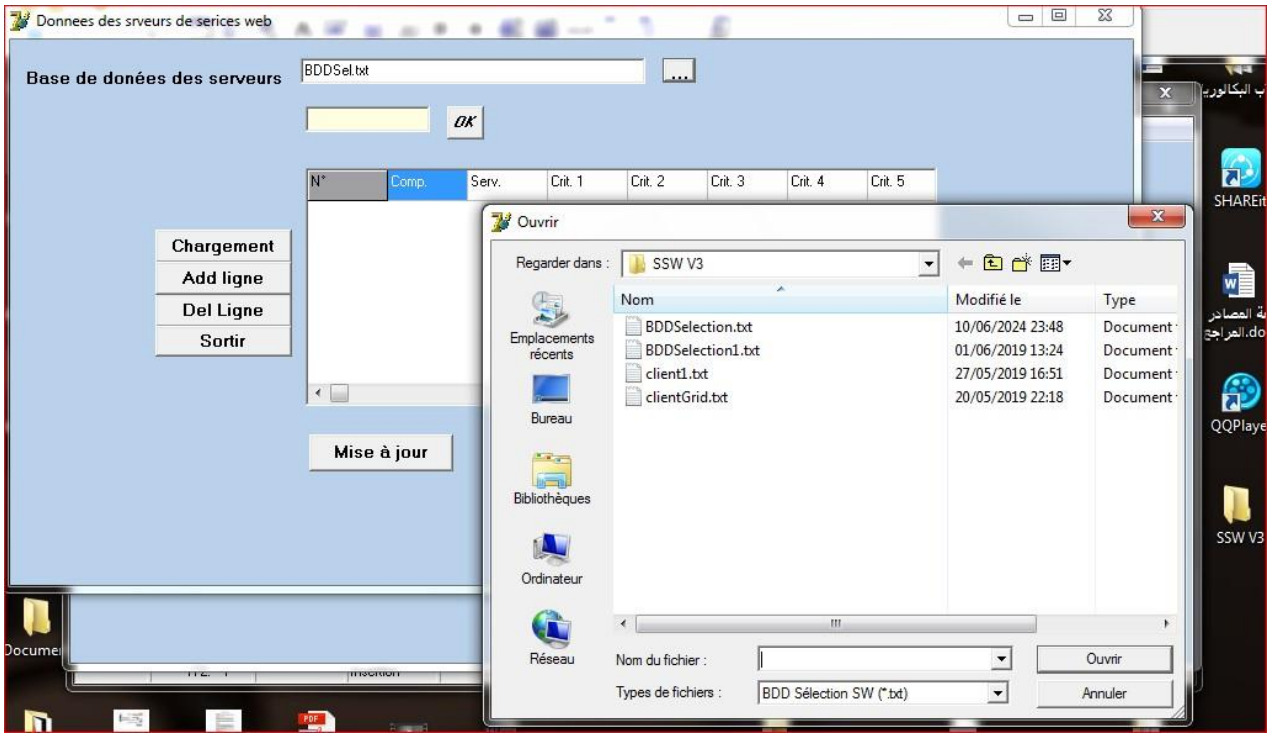


Figure V.8 : Choix d'une base de données.

Nous avons introduit les valeurs des critères proposés par plusieurs compagnies et nous avons chargé la BDD (figure V.9)

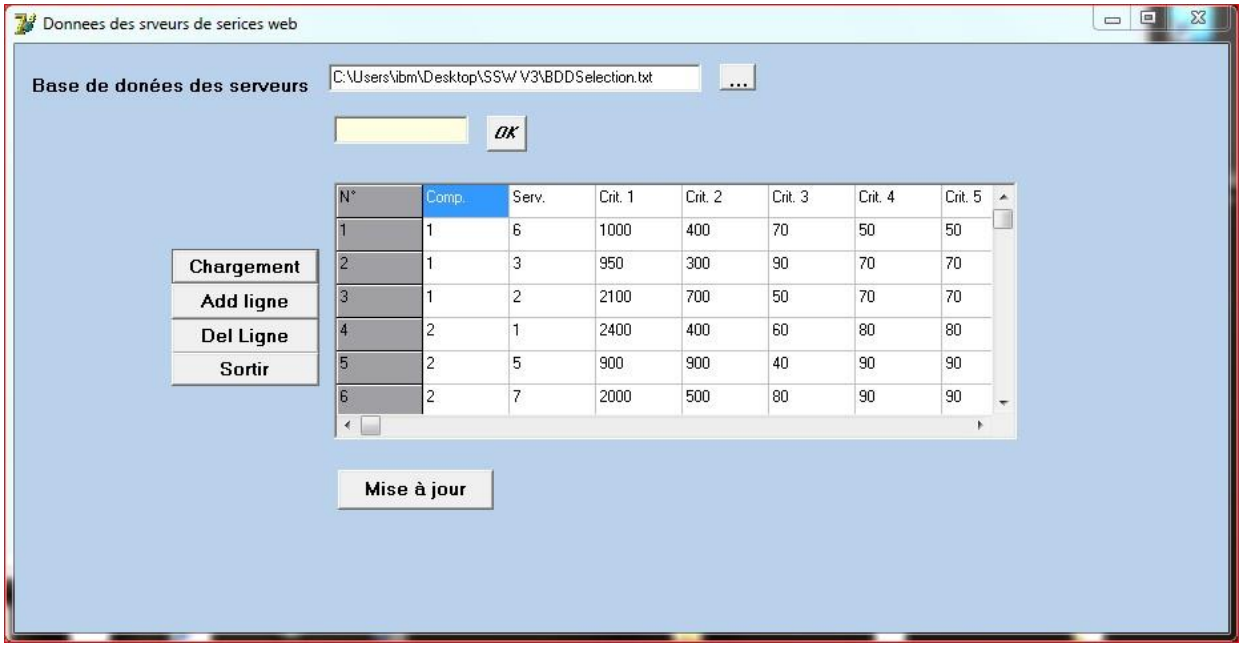


Figure V.9 : Chargement de la base de données.

La figure (V.10) et la figure (V.11) représentent le chargement des valeurs des critères des trois services exigés par le client.

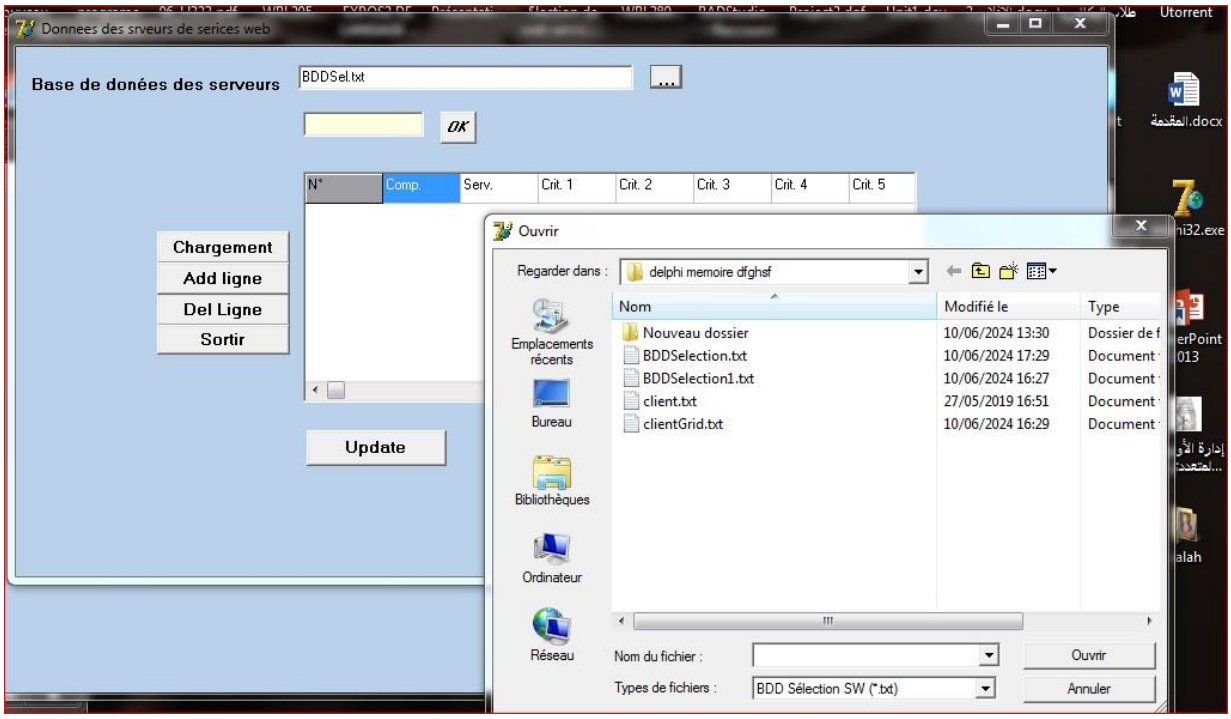


Figure V.10 : Choix du fichier C.txt.

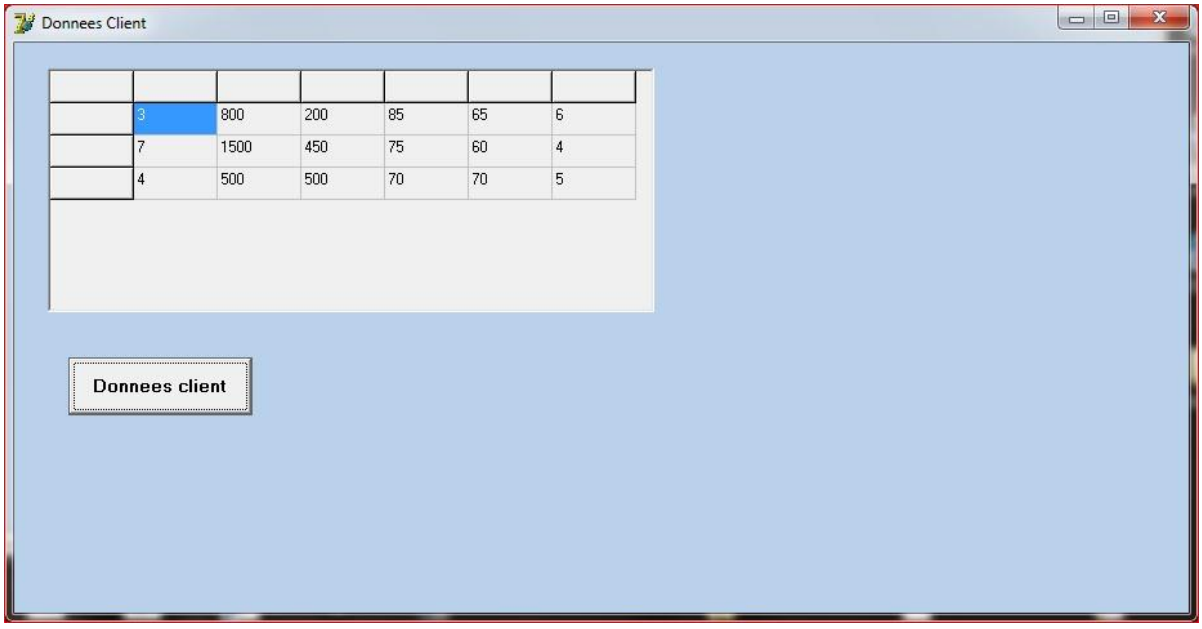


Figure V.11 : chargement des données client

La (figure V.12) représente le résultat de la sélection du service web optimal calculé via l'exécution de l'algorithme génétique.

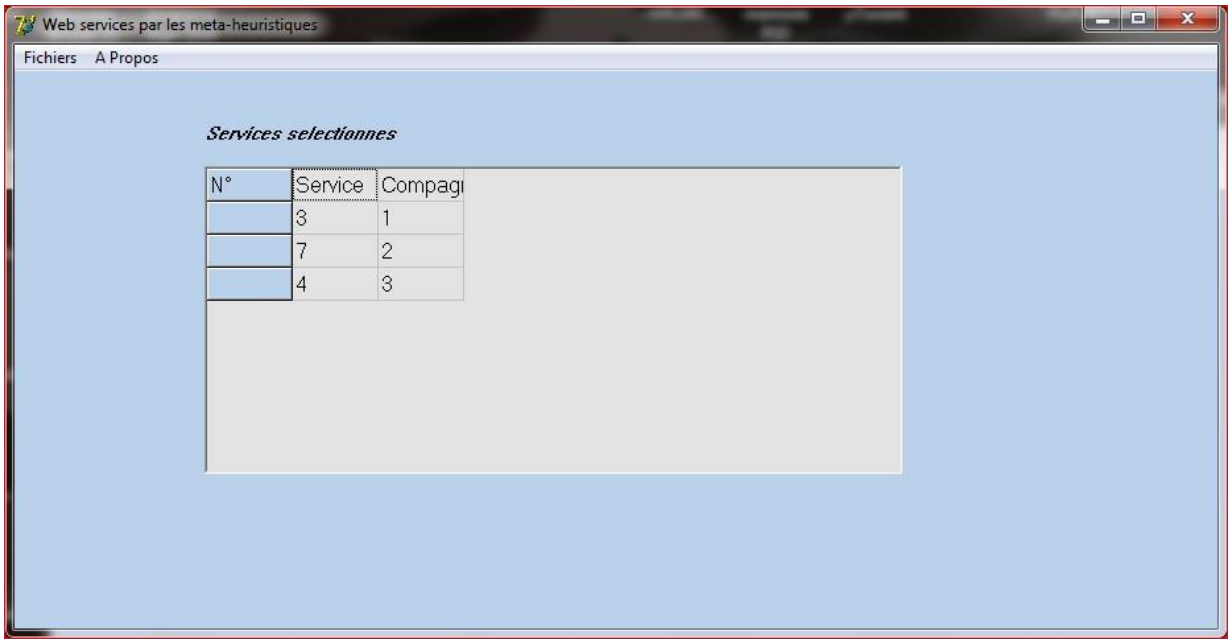


Figure V.12 : Résultat de la sélection.

5. Conclusion:

Dans ce chapitre nous avons présenté les résultats de la validation de notre approche, afin d'évaluer l'efficacité de ce dernier. Les résultats obtenus sont très acceptables et montrent qu'ils satisfassent les attentes de l'utilisateur. A partir des résultats expérimentaux obtenus précédemment. D'après les résultats qu'on a obtenus, on a confirmé l'efficacité d'algorithme recherche tabou.

Conclusion Générale

Conclusion Générale:

Ce projet de fin d'étude nous a permis d'enrichir et de perfectionner nos connaissances. Dans ce mémoire, nous avons présenté les technologies liées aux services Web et proposé un algorithme de sélection basé sur l'algorithme de Recherche tabou pour l'optimisation des compositions. Notre prototype sélectionne les compositions de services les plus satisfaisantes en se basant sur cinq critères de QoS : latence, coût, fiabilité, disponibilité et réputation.

La composition recherchée doit maximiser certains de ces critères positifs et minimiser d'autres critères négatifs, tout en satisfaisant un ensemble de contraintes globales. Pour améliorer les résultats de sélection, nous avons défini une nouvelle fonction objectif exprimée par des termes logarithmiques pour simplifier son calcul.

En perspective de ce travail, nous proposons d'implémenter d'autres algorithmes de sélection, tels que :

- La programmation par contrainte.
- La programmation dynamique.
- Les algorithmes d'optimisation basés sur Pareto.

Bibliographies

Bibliographies:

[1]: Bussler C "B2B Integration: Concepts and Architecture"; Springer-Verlag Berlin, 2003.

[2]: The Accredited Standards Committee (ASC) X12; Disponible sur: http://www.x12.org/

[3] : Rosetta Net ; Disponible à : http://www.rosettanet.org/

[4]:https://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.cics.ts.webservices.doc/ concepts/dfhws_definition.html (dernière consultation 22mars 2020) .

[5] : https://www.w3.org/TR/ws-arch/ (dernière consultation 22 mars 2020).

[6]:https://www.tutorialspoint.com/webservices/web_services_characteristics.htm (dernière consultation 23 mars 2020).

[7] : Valette M., Web services Communication inter langage, Version 2.0, Ecole supérieur d'Informatique de Paris, 8 mars 2006.

[8] : https://www.guru99.com/ web-service.html (dernière consultation 23 mars2020).

[9]:https://www.researchgate.net/figure/Web-Services.architecture_fig2_228781095 (dernière consultation 23 mars 2020)

[10] : https://www.w3schools.com/(dernière consultation 23 mars 2020)

[11]:https://www.irit.fr/journali3/hors_serie/annee2004/revue_i3_hs2004_01_07.pdf (dernière consultation 23 mars 2020)

[12]: J-M. Chauvet, Services Web aves SOAP, WSDL, UDDI, ebXML…, Edition EYROLLES

[13] : N. Mitra et Y. Lafon, SOAP Version 1.2 Part 0 : Primer (Second Edition 2003).

[14] : 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

[15] : A. Vezain, Les service web - présentation générale, rapport technique, Association HERMES, Février 2005.

[16] : M. Vialette, Web Services Communication inter langage, Version 2.0 Ecole Superieur d'Informatique de Paris, 8 mars 2006.

[17] : http://openclassrooms.com /coures/les-services-web

[18]: Benkhaled Hamid Naceur, REGRAGUI SidAhmed Tidjani,Selection de web
services a base de la qualité de service, 2016.

[**19] :**Benzina Y , Bekaddour H,La sélection multi objectifs des services web a base
de recuit simulé.

[20]: Hacene Halim, Etude comparative des méthodes heuristiques d'optimisation Combinatoire .

[21] : Sara MAQROT,Méthodes d'optimisation combinatoire en programmation mathématique. Application à la conception des systèmes de verger-maraîcher Thèse de doctorat, Université de Toulouse.

[22] Methodes exactes en optimisation combinatoire—principe fondamentale de la programation dynamique.

[23] : Dihya AISSAOUI, Souad BENHAMA,Méthode hybride pour la résolution des problèmes de programmation linéaire en nombres entiers,

Université Abderrahmane Mira, Béjaia. imisation combinatoire—principe fondamentale de la programation dynamique.

[24] : Djefel Rima,Kafi Sara, Une méthode basée méta-heuristique pour la sélection des services web, Université du 20 Août 1955- Skikda-.

[25] : Algorithme de colonie de fourmis wikipidia, Fr,wikipidia.org.

[26] : Conception de métaheuristiques d'optimisation pour la segmentation d'images,application aux images IRM du cerveau et aux images de tomographie par émission de positons ,Ahmed Nasreddine Benaichouche,TH2014EST 1106 archivage.pdf.

[27]: Recuit simulé : Définition, Exemple Fonctionnement, site :24 pm.com .

[28] : Kimouche Mohammed Amine,Les méthodes métaheuristiques pour l'optimisation en génie électrique,Université Mohamed Seddik Ben Yahia – Jijel,2019.

[29] : Delphi : définition et présentation de ce langage informatique journaldunet.fr