

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM



Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : Réseaux et Systèmes

Présenté par :
LAREF Amira

THEME :
Conception et réalisation d'un système de réservation
d'hébergement basé sur la sélection des services web

Soutenu le : 03/07/2022

Devant le jury composé de :

Mr	MICHAOUI Moulay Driss	MCA	Université de Mostaganem	Président
Mme	BENHAMED Siham	MAA	Université de Mostaganem	Examinatrice
Mme	HAMDANI Nesrine Djawida	MCB	ISTA-Oran	Encadrante
Mme	DEDDOUCHE Yamina	MAA	Université de Mostaganem	Co-Encadrante

Année Universitaire 2021-2022

Résumé

Les services Web sont des applications indépendantes, unifiées, distribuées et dynamiques qui peuvent être publiées, décrites, localisées ou connectées sur Internet pour créer des produits, des processus, etc. Ces applications peuvent être locales, distribuées ou basées sur le Web. Le fonctionnement des services Web dépend de la publication, de la recherche et de l'invocation des services, Ces fonctionnalités complexes peuvent être décomposées en un ensemble de fonctionnalités basique grâce à une architecture orientée service.

La qualité de services QoS est un ensemble de propriétés opérationnelles qui fournit un service conforme à des exigences en matière de temps de réponse, le coût, la fiabilité, la disponibilité, etc. Le choix du meilleur service parmi de nombreux services est la principale préoccupation de bon nombre des travaux présents dans la littérature, mais un seul service est incapable de répondre aux besoins de client donc il doit être combiné avec d'autres services afin d'obtenir les résultats souhaités, cela a conduit à l'apparence des services web composés.

L'objectif de ce travail est de trouver les meilleures compositions des services web et qui satisfaisant les besoins du client sur la base des critères de QoS. Pour cela, nous proposons une approche de sélection des services web qui combine une recherche locale basé sur la méthode de vote Condorcet avec un algorithme de recherche globale qui est l'Algorithme génétique.

Mots-clés : Services Web, Qualité de service, Sélection des services Web, composition des services web, Algorithme Génétique, Condorcet.

Abstract

Web services are independent, unified, distributed and dynamic applications that can be described, published, located or connected on the Internet to create products, processes, etc. These applications can be local, distributed, or web-based. The functioning of web services depends on the publication, the search and the invocation of services. These complex functionalities can be broken down into a basic set of functionalities thanks to a service-oriented architecture.

The quality-of-service QoS is a set of operational properties that provides a service conforming to requirements in terms of response time, cost, reliability, availability, etc. The choice of the best service among many services is the main concern of many of the works present in the literature, but a single service is unable to meet the client's needs so it must be combined with other services in order to achieve the desired results, this has led to the appearance of compound web services.

The objective of this work is to find the best compositions of web services that satisfy the needs of the client on the basis of QoS criteria. For this, we will propose an approach for the selection of web services which combines a local search based on the Condorcet voting method with a global search algorithm which is the Genetic Algorithm.

Keywords : Web service, Quality of service, selection of Web service, service web composition, Genetic Algorithm, Condorcet.

ملخص

خدمات الويب هي تطبيقات مستقلة وموحدة وموزعة وديناميكية يمكن نشرها أو وصفها أو تحديد موقعها أو توصيلها عبر الإنترنت لإنشاء منتجات وعمليات وما إلى ذلك. يمكن أن تكون هذه التطبيقات محلية أو موزعة أو قائمة على الويب. يعتمد عمل خدمات الويب على النشر والبحث واستدعاء الخدمات، ويمكن تقسيم هذه الوظائف المعقدة إلى مجموعة أساسية من الوظائف بفضل البنية الموجهة نحو الخدمة.

جودة الخدمة هي مجموعة من الخصائص التشغيلية التي توفر خدمة تلبية متطلبات وقت الاستجابة، والتكلفة، والموثوقية، والتوافر، وما إلى ذلك. ان اختيار أفضل خدمة من بين العديد من الخدمات هو الشغل الشاغل للعديد من الأعمال الموجودة في الأدبيات، ولكن خدمة واحدة فقط غير قادرة على تلبية احتياجات العميل لذلك يجب دمجها مع خدمات أخرى من أجل الحصول على النتائج المرجوة، وهذا ما أدى إلى ظهور خدمات الويب المركبة.

الهدف من هذا العمل هو العثور على أفضل تركيبات خدمات الويب التي تلبية احتياجات العميل على أساس معايير جودة الخدمة. لهذا، سنقترح نهجًا لاختيار خدمات الويب الذي يجمع بين البحث المحلي على أساس طريقة تصويت كوندورسي مع خوارزمية البحث العالمية وهي الخوارزمية الجينية.

كلمات مفتاحية

خدمات الويب، جودة الخدمة، اختيار خدمات الويب، تكوين خدمات الويب، الخوارزمية الجينية، تصويت كوندورسي.

Dédicaces

Je dédie ce modeste travail :

À Mes chères Parents,

À Mes frères, et belles-sœurs qui m'ont toujours soutenu

À Mes enfants Ilyes et Besma,

À Toute la famille,

À tous ceux qui m'aime et tous ceux que j'aime.

Et les plus adorables Lamia et Sarra

Amira

Remerciements

En premier lieu, je remercie le **Dieu** le tout puissant de m'avoir donné la santé, le courage et la volenté d'entamer et de terminer ce travail.

Tout d'abord, ce travail ne sera pas aussi riche et n'aurai pas pu avoir le jour sans l'aide de l'encadrement de Mme **HAMDANI** Nesrine Djawida et Mme **DEDDOUCHE** Yamina, je les remercie pour la qualité de leurs encadrement exceptionnelle, pour leurs patiences, leur rigueur et leurs disponibilités durant notre réalisation de ce mémoire.

Je remercie également tous les membres du jury Mr **MECHAOUI** Moulay Driss et Mme **BENHAMED** Sihem d'avoir bien voulu participer à l'évaluation de ce mémoire de fin d'étude.

Je tiens à remercier énormément Mr **ZAHMANI.M**, Mr **GHRICIA** et Mr **BADANI.M** et tous nos professeurs pour leurs aides et leurs conseils qui ont été d'un apport précieux pour l'accomplissement de ce travail.

Je tiens à exprimer ma profonde gratitude à mes parents pour leur soutien, leur confiance et leur patience, De plus, je tiens à remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet, car « un projet ne peut pas être le fruit d'une seule personne ».

A vous tous, je dis merci.

Liste des figures

Figure 1 – Schéma d'interactions entre les trois rôles [2]	5
Figure 2 – Structure d'un message SOAP [7]	7
Figure 3 – Pages de l'annuaire UDDI [11]	8
Figure 4 – Composition de services web [16]	10
Figure 5 – Cycle de vie d'un service web dans une composition [19]	11
Figure 6 – Stratégies de la sélection des SW [11]	14
Figure 7 – Sélection locale [2].....	15
Figure 8 – Sélection globale [2].....	15
Figure 9 – Classification des méthodes d'optimisation [23]	17
Figure 10 – Approches de résolution méta-heuristique [24]	18
Figure 11 – Architecture du système.	26
Figure 12 – Organigramme du système.	27
Figure 13 – Eléments principaux dans d'un algorithme génétique [32]	30
Figure 14 – Organigramme d'Algorithme génétique	30
Figure 15 – Exemple du codage réel d'un chromosome [35]	31
Figure 16 – Génération de la population initiale	32
Figure 17 – Croisement à un point [34]	35
Figure 18 – Croisement à multi points [34].....	35
Figure 19 – Exemple de croisement dans notre système	36
Figure 20 – Opérateur mutation [35]	36
Figure 21 – Diagramme de cas d'utilisation	38
Figure 22 – Diagramme de classe pour la BDD Voiture	39
Figure 23 – Diagramme de classes pour la BDD loisir.....	39
Figure 24 – Diagramme de classe pour la BDD hôtel	40
Figure 25 – Diagramme de classes de l'approche proposée	41
Figure 26 – Diagramme de séquence de l'approche proposée.....	44
Figure 27 – Code d'importation de la bibliothèque Gson	47
Figure 28 – Pages JSP de notre Application Web	48
Figure 29 – Code source pour créer nos fichiers JSON	50
Figure 30 – Interface Clever cloud pour hotel_db	51
Figure 31 – Chargement des matrices de QoS	52
Figure 32 – Classement de SW par ordre de préférence	53
Figure 33 – Résultats de Condorcet	53
Figure 34 – Population initiale de l'AG avec Condorcet	54
Figure 35 – Résultats AG + Condorcet	55
Figure 36 – Résultats AG	55
Figure 37 – Invocation des services Web.....	56
Figure 38 – Histogramme de comparaison de temps d'exécution	61
Figure 39 – Histogramme de comparaison de fitness	61

Liste des tableaux

Tableau 1 – Domaines de valeurs des critères [15]	9
Tableau 2 – Différentes approches relative à la sélection des SW	21
Tableau 3 – Exemple matrice de candidats	28
Tableau 4 – Exemple matrice de vote	28
Tableau 5 – Exemple matrice de choix	28
Tableau 6 – Exemple résultat Condorcet.	29
Tableau 7 – Les fonctions d'agrégation des critères de QoS [37].....	33
Tableau 8 – Résultats première configuration	57
Tableau 9 – Résultats deuxième configuration.....	57
Tableau 10 – Résultats de la troisième configuration	58
Tableau 11 – Résultats de la quatrième configuration.....	59
Tableau 12 – Résultats de la cinquième configuration	59
Tableau 13 – Représentation des maximums pour chaque population.....	60

Liste des équations

Equation	Titre	Page
1	Fonction d'agrégation du cout	33
2	Fonction d'agrégation du latence	33
3	Fonction d'agrégation du fiabilité	33
4	Fonction d'agrégation du disponibilité	33
5	Fonction d'agrégation du réputation	33
6	Fonction de qualité de service	33
7	Fonction de fitness	34

Liste des abréviations

Abbreviation	Expression Complète	page
W3C	World Wide Web Consortium	3
SOAP	Simple Object Access Protocol	4
HTTP	HyperText Transfer Protocol	4
XML	Extensible Markup Language	4
SOA	Service Oriented Architecture	4
SW	Service Web	4
URL	Uniform Resource Locator	5
WSDL	Web Services Description Language	7
UDDI	Universal Description Discovery and Integration	7
QoS	Quality of Service	8
BP	Business Process	11
NSGA II	Nondominated sorting genetic algorithm II	19
AG	Algorithmes Génétiques	19
FOA	Fruit Fly Optimization Algorithm	19
HS	Harmony Search	19
AMCD	Analyse Multicritères à la Décision	20
PaaS	Platform as a Service	47
Gson	Google Search Organization Network	47
JDBC	Java DataBase Connectivity	47
API	Application Programming Interface	48
SQL	Structured Query Language	48
SGBD	Système de Gestion de base de Données	48

JSP	Java Server Page	48
JSON	JavaScript Objet Notation	49
HTML	HyperText Markup Language	49

Table des matières

Introduction Générale	1
Chapitre 1 : Les services web.....	3
1.1 Introduction	3
1.2 Services web.....	3
1.3 Architecture orientée service SOA	4
1.4 Technologies de développements des SW	5
1.4.1 XML	6
1.4.2 HTTP	6
1.4.3 SOAP.....	6
1.4.4 WSDL.....	7
1.4.5 UDDI.....	7
1.5 Aspects fonctionnels et non fonctionnels des SW	8
1.6 Qualité de service QoS.....	8
1.7 Composition des services web.....	9
1.8 Types de composition des services web	10
1.9 Cycle de vie d'un SW dans une composition.....	11
1.10 Avantages des services web	11
1.11 Inconvénients des services web.....	12
1.12 Conclusion.....	12
Chapitre 2 : Sélection des services web	13
2.1 Introduction	13
2.2 Sélection des services web	13
2.3 Stratégies de sélection des SW	14
2.3.1 Sélection mono objectif.....	14
2.3.2 Sélection multi objectif.....	16
2.3.3 Sélection hybride.....	16
2.4 Problème d'Optimisation	16
2.4.1 Approches Heuristiques.....	17
2.4.2 Métaheuristique.....	17
2.5 Travaux réalisés	18
2.6 Discussion	23
2.7 Conclusion.....	24
Chapitre 3 : Conception du système	25

3.1	Introduction	25
3.2	Architecture du système	25
3.3	Organigramme du système	26
3.4	Condorcet	27
3.4.1	Fonctionnement de Condorcet	27
3.4.2	Avantages et inconvénients de Scrutin de Condorcet	29
3.5	Algorithme Génétique	29
3.5.1	Fonctionnement d'un algorithme génétique	29
3.5.2	Paramètres de dimensionnement d'un AG	37
3.5.3	Avantages d'algorithme génétique.....	37
3.5.4	Inconvénients d'algorithme génétique	38
3.6	Conception.....	38
3.6.1	Diagramme de cas d'utilisation	38
3.6.2	Diagrammes de classes.....	39
3.6.3	Diagramme de séquence.....	43
3.7	Conclusion.....	44
Chapitre 4 : Implémentation et expérimentations.....		46
4.1	Introduction	46
4.2	Environnement de développement	46
4.2.1	Environnement Matériel.....	46
4.2.2	Environnement Logiciel	46
4.2.3	Langages de programmations	48
4.2.4	Description de la BDD	50
4.3	Présentation des interfaces	51
4.4	Expérimentation	56
4.5	Conclusion	62
Conclusion Générale		63
Bibliographies.....		65

Introduction Générale

Les applications Web contribuent grandement à la sélection des services Web qui conviennent aux exigences de l'utilisateur parmi les nombreux services offerts, tels que le choix de la bonne destination, de la réservation d'hôtel, la location de voiture, etc. De nombreux travaux et recherches ont été proposés pour la sélection des services web qui est basée généralement sur les exigences fonctionnelles du client, mais ces services peuvent ne pas être de la même qualité que celle à laquelle l'utilisateur s'attend et pour cette raison, la sélection du service Web est basée principalement sur la qualité de service (QoS) qui satisfait les besoins déclarés. Ces derniers peuvent être liés à : l'accessibilité, la disponibilité, le temps de réponse, le coût, la fiabilité, etc.

Une simple recherche sur internet permet de trouver de nombreux services Web répondant à ces critères, mais lequel de ces services sera le plus pratique, rapide, disponible, fiable et avec la même qualité qu'on attend de lui ?

Le but de notre travail est de réaliser un système de réservation d'hébergement basé sur la sélection des services web. L'application doit fournir à l'utilisateur un ensemble de services web composés configurés et adaptés à ses besoins afin d'invoquer les différents établissements de réservation d'hôtel, loisir et location de voiture. La composition de Web services spécifie quels services ont besoin d'être invoqués, dans quel ordre et comment gérer les conditions d'interaction.

L'objectif principal de ce travail est de choisir et retourner la meilleure composition de services web pour la réservation d'hébergement. Pour cela, Nous proposons une sélection hybride qui combine une recherche locale basée sur la méthode de vote Condorcet avec une recherche globale à base d'un algorithme génétique pour obtenir une solution appropriée à notre problème.

Plan du manuscrit

Outre le résumé, l'introduction et la conclusion générale, ce Projet de fin d'étude est organisé en quatre chapitres, comme suit :

- **Chapitre 1** : Les services Web

Ce chapitre est consacré à la définition des Services Web, de l'architecture orientée service SOA et les différentes technologies de développement. Nous avons aussi parlé du fonctionnement des SW et nous avons cité leurs propriétés fonctionnelles et non fonctionnelles et les critères de la qualité de services QoS, sans oublier la composition des services web qui nous permet de sélectionner le service web composite le plus approprié parmi l'ensemble des services existants.

- **Chapitre 2** : La sélection des services web

Dans ce chapitre, Nous allons présenter un *état de l'art* sur la sélection des services web. Commençons par la définition des différentes stratégies de sélection qui existent et nous avons mis la lumière sur le problème d'optimisation et nous avons terminé ce chapitre par une description détaillée de quelques travaux relatifs à la sélection de services web.

- **Chapitre 3** : Conception de notre système

Ce chapitre décrit une conception détaillée de notre système, son architecture et les concepts de base de l'approche proposée pour la sélection des services web composite qui combine une méthode de recherche locale basée sur le vote de Condorcet avec un algorithme de recherche globale qui est l'Algorithme Génétique.

- **Chapitre 4** : Implémentation et Expérimentations

Le dernier chapitre présente l'environnement de développement, l'implémentation de notre système, expérimentations et une discussion des résultats obtenus.

Ce mémoire contient aussi une liste des figures, tableaux, une table des abréviations, une table d'équations et une table de matières avant l'introduction générale et une bibliographie à la fin du manuscrit.

Chapitre 1

Les services web

1.1 Introduction

Ces dernières années, le web a connu une grande popularité et un grand succès en permettant une interaction entre les êtres humains et les machines grâce à des services web qui sont utilisée par des applications disponibles sur internet et qui permettent à chacun d'entre eux d'effectuer une tâche précise.

Dans ce chapitre, nous abordons les services Web, l'architecture orientée services, les technologies de développement ainsi que le fonctionnement des services web. De plus, nous allons connaître les propriétés d'un service web (fonctionnelles et non-fonctionnelles) et la qualité de service et nous terminions ce chapitre par quelques avantages et inconvénients des services web.

1.2 Services web

Est une technologie qui met à disposition un service via internet. Il permet à deux ou plusieurs machines (client/serveur) de communiquer entre eux à distance (via le réseau) indépendamment de leurs plateformes d'exécution et langages de programmation.

Plusieurs définitions des services Web ont été proposées dans la littérature, notamment :

Le Groupe W3C (World Wide Web Consortium) [1] :

« Un service Web est un système conçu pour permettre l'interopérabilité des applications à travers un réseau. Il est caractérisé par un format de description interprétable/compréhensible automatiquement par la machine (en particulier WSDL).

D'autres systèmes peuvent interagir avec le service Web selon la manière prescrite dans sa description et en utilisant des messages SOAP, généralement transmis via le protocole HTTP et sérialisés en XML et en d'autres standards du Web ».

Selon IBM [2] :

« Les services web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto descriptives qui peuvent être publiées, localisées et invoquées depuis le web. Les services web effectuent des actions allant de simples requêtes à des processus complexes. Une fois qu'un service Web est déployé, d'autres applications peuvent le découvrir et l'invoquer ».

De ces définitions, on peut déduire quelques points forts des services web notamment :

- Une interface écrite avec un langage de description standard et lisible ;
- Un service web est faiblement couplé ;
- Il utilise des protocoles indépendants de leurs plateformes d'exécution ;
- Il utilise les standards HTML (HyperText Transfer Protocole) et XML (eXtensible Markup Language) comme des moyens pour l'échange des données ;
- Il améliore l'interopérabilité grâce à l'utilisation des normes web existantes tel que SOAP (Simple Object Access Protocol).

1.3 Architecture orientée service SOA

Il s'agit d'une architecture logicielle qui permet un déploiement et une gestion indépendants de plusieurs composants (service web) pour communiquer et travailler ensemble sous la forme d'une application logicielle commune pour différents systèmes, et pour cela le modèle de conception SOA permet de réaliser des applications entièrement distribuées.

En offrant des services sur des protocoles réseau standards tel que SOAP, SOA afin de mettre en œuvre une intégration à partir de zéro et permet de réutiliser les fonctionnalités existantes au lieu de les créer [3]. Comme illustre la Figure, l'architecture de base et le fonctionnement des SW (service web) comprend trois rôles principaux :

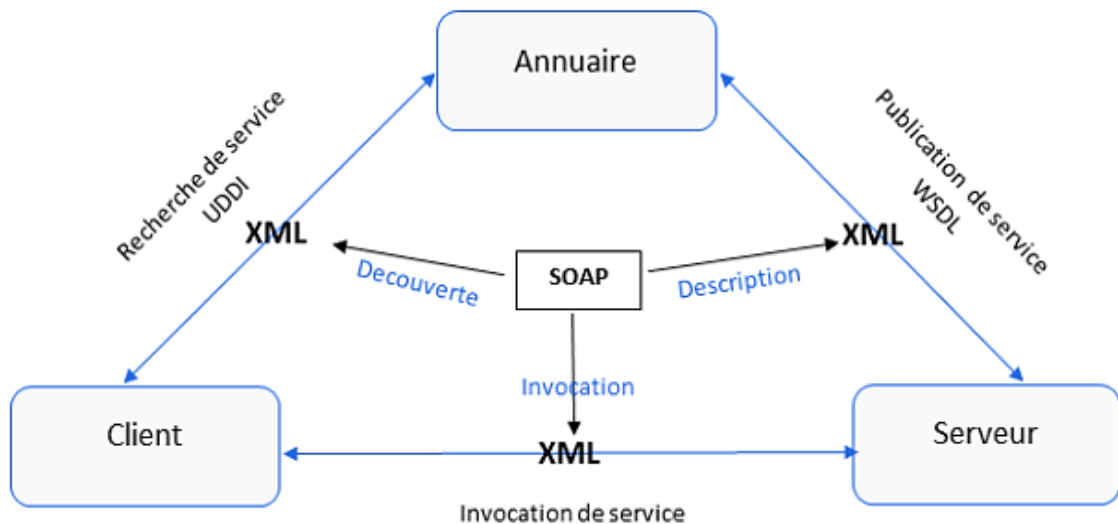


Figure 1 – Schéma d'interactions entre les trois rôles [2]

- **Fournisseur (serveur) de service** : c'est le propriétaire de services, il crée les SW qu'il fournit dans le registre des services, ainsi il est responsable des conditions d'utilisation du service. Le fournisseur peut être un client pour un autre fournisseur.
- **L'annuaire (registre) de service** : c'est une base de données qui contient les URLs (Uniform Resource Locator) et les descriptions des services créés par le fournisseur, de plus, il est responsable de fournir des informations sur le service au client.
- **Le client (demandeur) de service** : c'est le consommateur de service (application cliente), il cherche un service dans l'annuaire. Une fois trouvé le service il récupère l'URL et la description de ce service puis contacte le fournisseur pour obtenir le service demandé [3].

1.4 Technologies de développements des SW

Les services Web reposent dans leur fonctionnement sur l'utilisation d'un ensemble de technologies qui partagent une structure commune et contribuent au développement et à l'amélioration des SW. Dans cette section, nous présenterons les principales technologies standards de base qui sont basées sur le langage XML ainsi que le protocole de transport HTTP.

1.4.1 XML

Est une norme pour la représentation des données standardisée par W3C en 1998. Il facilite l'échange d'information sur l'internet, son objectif est de formaliser des données textuelles facile à lire et à comprendre. Ainsi, l'XML est également très accessible car il ne nécessite pas un logiciel d'édition de code pour l'écrire. Les SW utilisent l'XML car ils sont universels, extensibles, adoptés par tous les acteurs de l'internet et permettent une extensibilité facile grâce à l'utilisation d'espaces de noms [4].

1.4.2 HTTP

HTTP est un protocole de communication inventé par Tim Berners-Lee dans les années 1990 qui crée une liaison entre un client (ordinateur) et un serveur web. Une fois le client envoie une demande (requête) au serveur à travers un navigateur web, le serveur apporte une réponse presque instantanée. Autrement dit, le HTTP nous permettons d'accéder à un contenu en ordonnant au serveur d'effectuer une action [5]. La requête HTTP est identifiée par une URL qui est composée de [6] :

- Le protocole (HTTP) ;
- L'hôte (host) ;
- Le numéro de port (80, 8080, etc.) ;
- Le chemin qui mène à la ressource sur le serveur (Path) ;
- La ressource demandée par l'URL.

1.4.3 SOAP

Est un protocole de communication utilisé sur le réseau (internet) proposé par le W3C, il permet l'échange des informations entre les applications et l'accès aux SW. L'utilisation du protocole SOAP favorise l'interopérabilité entre les applications à travers le web. SOAP intervient lorsqu'un système doit accéder à un autre système de façon ordonnée et limitée. Il est basé sur les deux normes : XML pour la structure des messages et HTTP pour le transport de données. L'utilisation de HTTP lui permet de passer par des proxys et pare-feu.

Les messages SOAP sont organisés dans un document XML et contient deux éléments obligatoires : L'enveloppe qui contient des métadonnées et un corps (body) de message qui contient les données à proprement parler. Le SOAP Header et SOAP Fault sont des blocs optionnels, le premier contient des informations d'en-têtes sur le message et le Fault Utilisé

pour signaler les erreurs lors du traitement des messages. Le protocole header contient l'entête du protocole de transport de données (HTTP). La Figure 2 montre la structure d'un message SOAP :

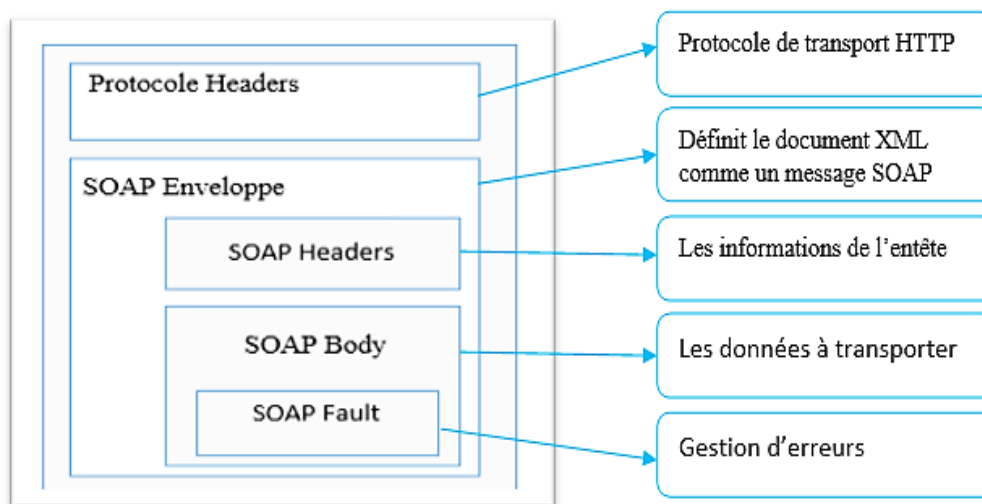


Figure 2 – Structure d'un message SOAP [7]

1.4.4 WSDL

Web Services Description Language, est un langage standard de description qui peut être utilisé pour décrire les fonctions, les données, les types de données et les protocoles d'échange proposés par un service web. C'est une recommandation du W3C. Un client appelant un SW peut lire WSDL pour déterminer quelles fonctions sont disponibles sur le serveur. Tous les types de données spéciaux utilisés sont intégrés dans le fichier WSDL sous forme XML [8] [9].

1.4.5 UDDI

L'annuaire UDDI (Universal Description Discovery and Integration) est une norme de découverte et de publication d'informations sur les SW. La spécification UDDI vise à créer un espace de travail indépendant de la plate-forme (Framework) pour décrire, découvrir et intégrer des services d'entreprises.

L'entrée d'annuaire UDDI consiste en un fichier XML décrivant l'entreprise et les services qu'elle fournit. Chaque entrée de registre UDDI se compose de trois parties [10], comme illustré à la figure 3 :

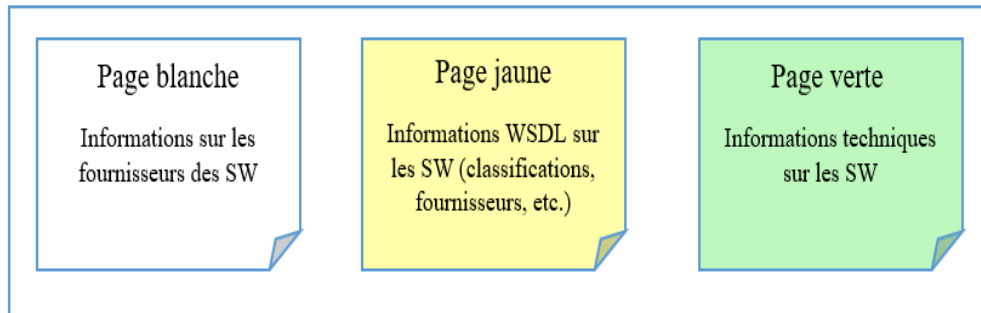


Figure 3 – Pages de l'annuaire UDDI [11]

Les descriptions des SW publiées par les fournisseurs en UDDI sont sous forme de fichiers WSDL, ainsi les clients peuvent trouver plus facilement les SW dont ils ont besoin en interrogeant l'annuaire UDDI.

1.5 Aspects fonctionnels et non fonctionnels des SW

Les SW sont définis par leurs propriétés fonctionnelles et non fonctionnelles. En effet, ils sont caractérisés par les fonctionnalités qu'ils peuvent fournir, si deux services offrent les mêmes fonctionnalités ils se distinguent par leurs propriétés non fonctionnelles.

- **Propriétés fonctionnelles** : Définit le comportement global de service web. Autrement dit, tous les détails sur la façon dont le service est invoqué, où il se trouve, la syntaxe du message, les protocoles de réseaux pour transporter ces messages, etc. La représentation des services Web, les méthodes de recherche, et les critères de sélection proposés aux clients sont proposés par les registres publics UDDI [12].
- **Propriétés non fonctionnelles** : Représente la capacité avec laquelle un service Web peut fonctionner normalement dans des conditions appropriées en fonction des attributs de la qualité de services [13].

1.6 Qualité de service QoS

Est un ensemble de caractéristiques opérationnelles d'un service qui doivent être prises en compte dans la réalisation du service pour répondre aux besoins énoncés [14]. Ces besoins peuvent être liés à : la disponibilité, le coût, le temps de réponse, la fiabilité, etc. Ces

paramètres sont des critères de choix lorsqu'on sélectionne parmi plusieurs services. Dans cette section nous présentons quelques attributs génériques de la QoS [2] :

- **Le coût** : c'est le frais de service, il est affecté par la valeur de la fonctionnalité ;
- **Le temps de réponse (latence)** : Le temps entre l'envoi de la requête et la réception des résultats ;
- **La fiabilité** : La capacité du service à exécuter correctement ses fonctions ;
- **La disponibilité** : La possibilité d'accéder au service ;
- **La réputation** : C'est une mesure de fidélité. Il est principalement basé sur les expériences des utilisateurs (commentaires) lors de l'utilisation de ce service ;
- **La sécurité** : C'est un ensemble d'attributs (confidentialité, chiffrement des messages, contrôle d'accès, etc.) ;
- **La scalabilité** : La capacité du service à traiter autant d'opérations que possible pendant une période de temps donnée tout en conservant les mêmes performances ;
- **Le débit** : Nombre de demandes (requêtes) soumises par unité de temps ;

Les domaines de valeur de quelques normes de qualité de service sont censés être définis comme suit dans le tableau 1 :

Tableau 1 – Domaines de valeurs des critères [15]

Critères	Domaine de valeurs
Coût	1 – 30 euros
Temps de réponse	1 – 300 ms
Fiabilité	0 – 5 %
Disponibilité	0 – 1 %
Réputation	0 – 1

1.7 Composition des services web

La communication entre le client et le fournisseur de services est établie via le réseau (internet) où le client veut satisfaire leurs demandes à travers un ensemble de fonctionnalités

en tant que service. Un SW assure l'exécution d'une seule fonctionnalité ce qui le rend incapable de répondre aux exigences du client donc il doit être lié et combiné avec d'autres SW pour donner les résultats entendus. Cela a conduit à l'apparence du service web composite ou la composition des services web.

La composition des SW est le mécanisme qui permet à l'intégration des services existants de créer un nouveau SW à valeur ajoutée (Figure 4). Elle définit aussi les services à invoquer et dans quel ordre.

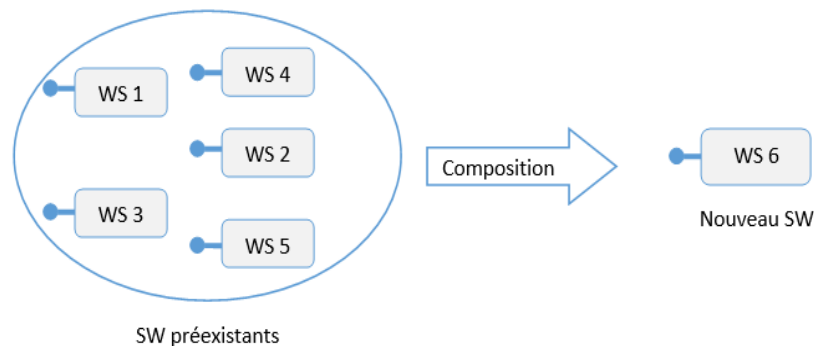


Figure 4 – Composition de services web [16]

1.8 Types de composition des services web

Il existe trois types de composition :

- **La composition manuelle :** c'est l'utilisateur qui génère la composition des SW via un éditeur de texte à la main sans l'utilisation d'un outil dédié.
- **La composition semi-automatique :** Il s'agit d'une étape avancée par rapport à la configuration manuelle, elle fournit des suggestions sémantiques pour aider à sélectionner les SW dans le processus de composition.
- **La composition automatique :** est entièrement automatisé, elle prend en charge l'ensemble du processus de composition et l'exécute automatiquement, sans intervention de l'utilisateur.

1.9 Cycle de vie d'un SW dans une composition

- **Découverte de service web** : est l'étape la plus importante car le client doit chercher le service ou le processus métier qui répond à ses besoins avant d'interagir avec le fournisseur de ce service [17].
- **Sélection du service web** : consiste à sélectionner le SW le plus adéquat aux exigences de l'utilisateur parmi tous les services web disponibles [18].
- **Composition de service web** : est la combinaison et l'exécution des SW avec d'autres service afin d'accomplir une certaine tâche complexe.

La figure 5 illustre le cycle de vie d'un service web dans une composition qui commence par la spécification des besoins de l'utilisateur (défini les propriétés fonctionnelles et non fonctionnelles QoS), ces derniers sont recherchés dans l'annuaire à l'aide de l'utilisation des algorithmes de recherche, cette recherche nous donne un processus métier (BP) qui comprend un ensemble de tâches abstraites [19].

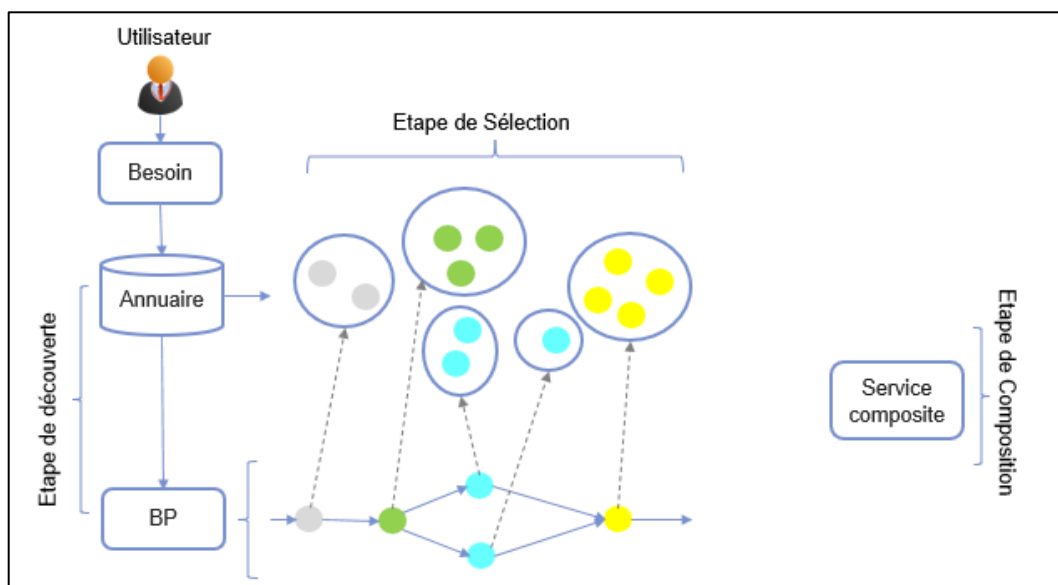


Figure 5 – Cycle de vie d'un service web dans une composition [19]

1.10 Avantages des services web

- Les services web permettent une meilleure interopérabilité entre les logiciels ;
- Permettent la communication entre les programmes sans tenir en compte leurs langage de programmation et plates formes d'exécutions ;

- Ils fonctionnent malgré les pare-feux grâce au protocole THHP ;
- Facilite l'échange de données est la compréhension de fonctionnement grâce à l'XML qui formalise des données textuelles facile à lire et à comprendre ;
- Un système faiblement couplé.

1.11 Inconvénients des services web

La technologie SW présente de nombreux inconvénients, notamment [20] :

- Les services Web utilisent XML qui produit de gros paquets de données, ce qui pose des problèmes si la connexion réseau est lente ;
- Les mesures de sécurité imposées par les pare-feux sont facilement contournées ;
- Il est difficile de garantir la fiabilité.

1.12 Conclusion

Nous avons dans ce chapitre, défini d'une façon générale qu'est-ce que c'est un service web, son principe de fonctionnement et les différentes technologies de développements tel que SOAP, WSDL, UDDI, etc. ainsi la QoS qui fournit un ensemble complet de critères qui répondent aux besoins des utilisateurs et de fournisseurs de service selon les ressources réseau disponibles.

Un service web unique ne peut pas satisfaire les besoins déclarés ce que nécessite une composition des SW qui combine plusieurs services pour produire un service plus complexe pour répondre à une demande d'utilisateur. Nous avons vu aussi dans cette section la définition et les types de la composition des SW, ainsi le cycle de vie d'un SW dans une composition. Le chapitre suivant est un état de l'art dans lequel nous allons présenter plus en détail la sélection et l'optimisation des services web ainsi qu'une description de certains travaux liés à la sélection des SW.

Chapitre 2

Sélection des services web

2.1 Introduction

Aujourd'hui, un nombre important de services web sont disponibles en ligne. Grâce à cette augmentation, de nombreux problèmes sont apparus, notamment liés à la sélection, la disponibilité, la qualité de service, etc.

Dans ce chapitre nous définissons le principe de la sélection des SW, en présentant les stratégies de classification des méthodes de sélection basées sur la QoS qui existent actuellement ainsi l'optimisation. Ensuite, Nous passerons en revue les principaux travaux proposés liés à la métaheuristique.

2.2 Sélection des services web

La croissance rapide du nombre de services Web offrant des services similaires de même qualité a incité les fournisseurs de services à concentrer leurs attentions sur les aspects non fonctionnelles des SW plus précisément la QoS afin d'améliorer la qualité de leurs services et facilite la sélection.

Selon [13] La sélection de SW consiste à sélectionner le service le plus adéquat selon un besoin. La sélection se base principalement sur la description de service qui est une interface de service et qui identifie un ensemble d'opérations et un ensemble de propriétés pour qualifier le service.

2.3 Stratégies de sélection des SW

Il existe trois grandes classes : La sélection mono objective, la sélection multi objectif et la sélection hybride. La Figure 6 montre la hiérarchie de ces classes :

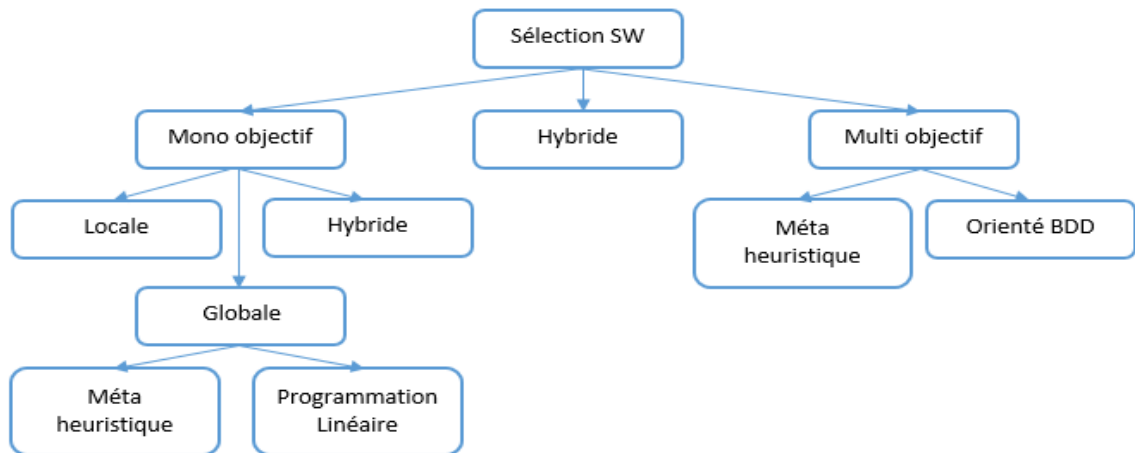


Figure 6 – Stratégies de la sélection des SW [11]

2.3.1 Sélection mono objectif

Cette stratégie de sélection transforme les objectifs du problème d'optimisation en une seule fonction objectif afin de déterminer une solution optimale, elle est composée de deux stratégies principales : la sélection locale et la sélection globale. La combinaison des deux est connue sous le nom hybride.

2.3.1.1 Sélection locale

Son objectif est de sélectionner individuellement le meilleur SW pour chaque tâche en considérant les contraintes QoS de chaque tâche plutôt que de considérer les contraintes QoS globales exprimées pour toutes les tâches. La Figure 7 décrit la stratégie de sélection locale [2].

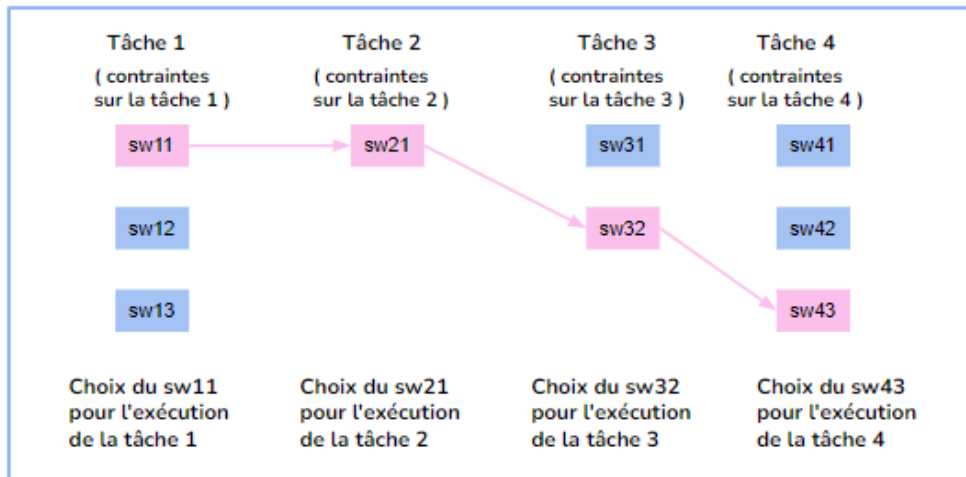


Figure 7 – Sélection locale [2]

2.3.1.2 Sélection globale

Contrairement à la méthode précédente, un ensemble de SW est choisi [21], Cela garantit la meilleure qualité globale en tenant compte des contraintes de la QoS et des préférences globales définies pour toutes les tâches. La Figure 8 décrit la stratégie de sélection globale.

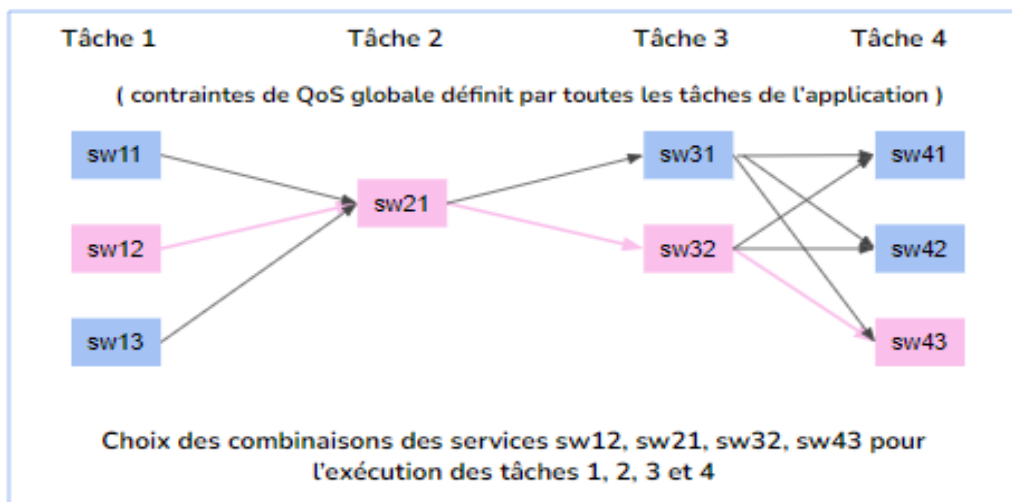


Figure 8 – Sélection globale [2]

2.3.1.3 Sélection Hybride

La sélection hybride combine les deux classes précédentes. Selon [15] la recherche commence par une optimisation globale après continue à travailler avec une optimisation locale.

2.3.2 Sélection multi objectif

Contrairement à la sélection mono objectif la sélection multi objectif nous fournit un ensemble de solutions optimales. Elle est divisée en deux classes principales la sélection métaheuristique et la sélection orienté bases de données qui traite un problème multi-objectif localement ou globalement sans le réduire à un problème mono objectif, elle repose sur le principe des requêtes Top-k ou Skyline qui sont basés sur la dominance floue et qui ont pour objectif de réduire l'espace de recherche et renvoie les points les plus pertinents en choisissant parmi les incomparables [2].

2.3.3 Sélection hybride

Cette approche combine les deux approches précédentes (mono objectif et multi objectif). Elle commence par une recherche multi objectif pour plusieurs classes, puis regroupe les services de chaque classe et choisi des représentants de groupe, ensuite elle continu avec une recherche mono objectif sur les représentants de ces groupe [22].

2.4 Problème d'Optimisation

L'optimisation est une approche des mathématiques et de l'informatique dans le but de modéliser et d'analyser des problèmes réels et les résoudre de manière analytique. Elle consiste à identifier des solutions permettant d'atteindre un objectif quantitatif tout en respectant des contraintes potentielles. Un problème d'optimisation est un modèle formel (mathématique) utilisé pour résoudre un problème réel et qui cherche à maximiser ou minimiser une fonction objectif sous des contraintes [23].

Comme le montre la figure 9, il existe de nombreuses méthodes d'optimisation. Le principe d'une méthode exacte est de trouver des solutions dont l'optimalité est garantie pour des problèmes de grande taille, mais nous pouvons rechercher des solutions de bonne qualité sans garantie avec un temps optimal plus réduit grâce à l'utilisation des méthodes heuristiques et méta heuristiques.

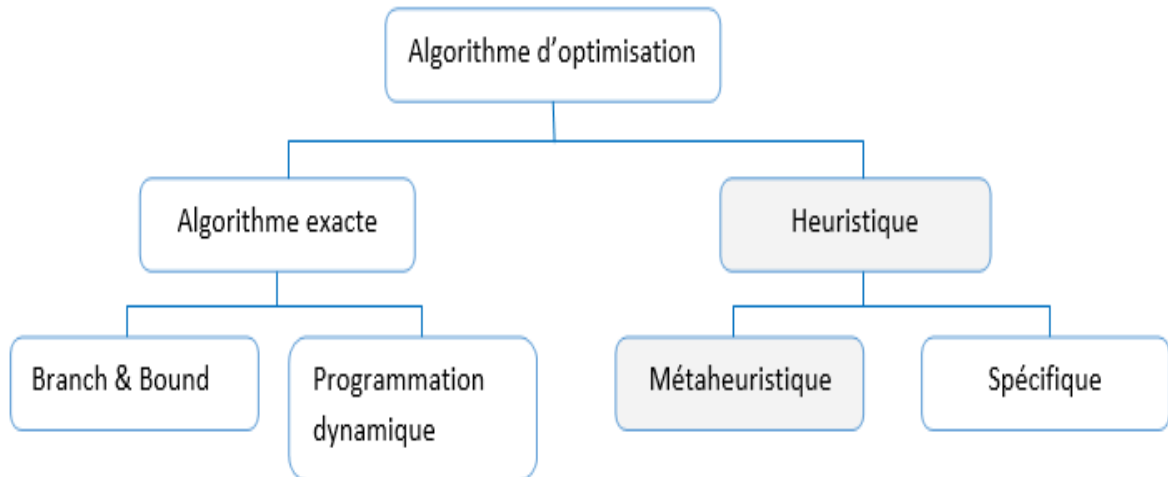


Figure 9 – Classification des méthodes d'optimisation [23]

2.4.1 Approches Heuristiques

Les heuristiques sont des méthodes de résolution algorithmique qui permettent de trouver rapidement des solutions raisonnables à tout problème de prise de décision sans faire une analyse détaillée ou complète. Plus précisément, sont des stratégies qui se déplacent intelligemment dans un espace de solutions jusqu'à obtenir une meilleure solution approchée dans un temps raisonnable en s'appuyant sur similitudes avec des problèmes déjà abordés afin de ne garder qu'une série limitée de solutions pour aller vers la solution optimale.

2.4.2 Métaheuristique

Sont des algorithmes stochastiques itératifs visant à résoudre des problèmes d'optimisation pour lesquels il n'existe pas de méthode classique adaptée. Les métaheuristique combinent généralement plusieurs approches heuristiques, pour cela, ils sont non déterministes et ne donnent aucune garantie d'optimalité mais ils cherchent de trouver rapidement les meilleures solutions (raisonnable et proche de la solution optimale) dans un espace de recherche [23].

Les méta-heuristique sont classés selon le nombre de solutions : des méta-heuristiques à solution unique et méta-heuristiques à population de solutions comme illustre la figure 10 :

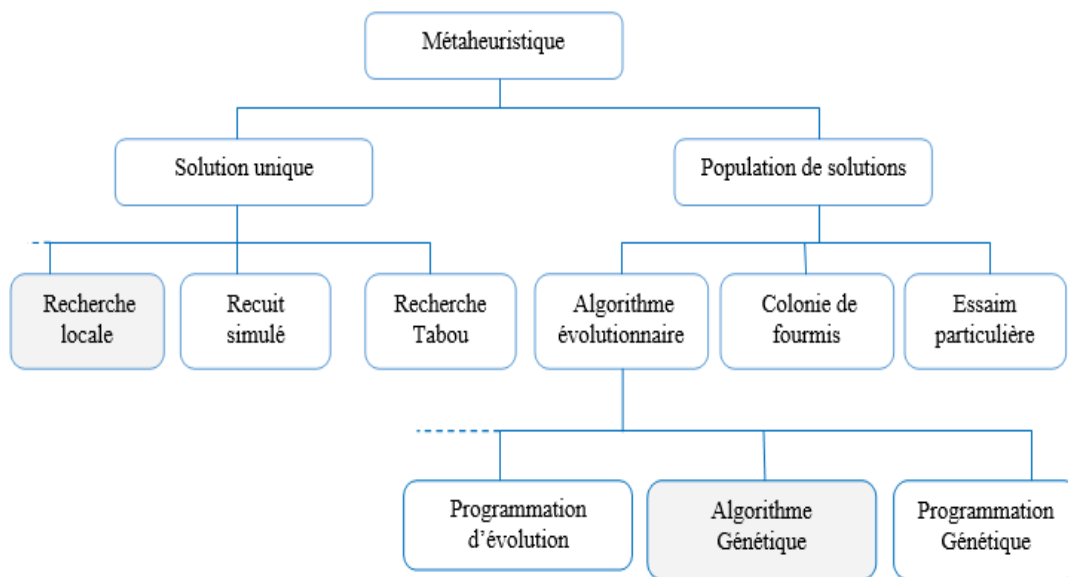


Figure 10 – Approches de résolution méta-heuristique [24]

2.4.2.1 Métaheuristiques à base de solution unique

Cette classe commence la recherche avec une seule solution, Ensuite, elle l'améliore grâce à une série d'itérations basée sur le concept de voisinage. Nous trouvons dans cette section plusieurs familles d'algorithmes de recherche tel que : Recherche Tabou, Recuit simulé, etc. [15].

2.4.2.2 Métaheuristiques à base de population de solutions

Contrairement à l'approche à solution unique, cette classe commence la recherche par un ensemble de solutions pour obtenir la solution la plus optimale du problème donné. Cette classe est divisée en plusieurs familles d'algorithmes de recherche tel que : Algorithme évolutionnaires, essais particuliers, etc. [15].

2.5 Travaux réalisés

En raison de l'importance de la sélection du meilleur SW composite, de nombreuses études et recherches scientifiques ont été traitées ces dernières années. Dans cette section nous présentons quelques travaux déjà soumis liés aux métaheuristiques dans la sélection globale, locale et hybride et qui ont retenu notre attention.

Dans l'article de [25], L'auteur a utilisé un algorithme génétique élitiste de tri Non-dominé NSGA II. Cet algorithme a été considéré comme un algorithme standard pour trouver l'ensemble de Pareto optimal (la satisfaction des individus) avec variété de solutions. Selon les résultats donnés, NSGA II est capable de donner un front de Pareto acceptable mais si la taille de la population dépasse une certaine limite le temps de reponse sera plus raisonnable.

Un autre algorithme d'optimisation a été proposé par [22] c'était l'Algorithme Mimétique. Cet algorithme est un Algorithme Génétique (GA) hybridé avec un algorithme de recherche locale, où l'auteur a appliqué la recherche locale pour chaque nouvel individu créé (dans l'GA) pendant un nombre d'itérations définit pour trouver et renvoie les meilleurs individus qui remplaceront les mauvais individus

Loin de l'algorithme génétique, une autre famille de métaheuristiques d'optimisation basée sur un mouvement par essaims a été proposé par [15]. L'algorithme FOA « *Fruit Fly Optimization Algorithm* » a été utilisé pour trouver une optimisation globale d'une composition des SW. L'auteur a appliqué cet algorithme dont le but est de choisir le meilleur déplacement et qui représente la meilleure composition.

[11] Ont proposé une autre méthode pour sélectionner les meilleures compositions de SW en termes de QoS et qui se base sur les comportements des colonies d'abeilles. « *Bees Algorithm* » ou L'Algorithme d'Abeilles est basé sur la population de recherche, il effectue une recherche de voisinage avec une recherche aléatoire et c'est sur quoi l'auteur s'est appuyé pour résoudre les problèmes d'optimisation. Les résultats analysés ont montré l'efficacité de l'algorithme par rapport à certains autres, mais cela ne cache pas la possibilité de ne pas atteindre l'objectif en raison du mauvais point de départ car l'algorithme commence par une population initiale aléatoire et il tente d'améliorer par suite en passant par plusieurs étapes.

Dans le travail intitulé « *la sélection multi objectifs des services web de Recuit Simulé* » [26], les auteurs ont utilisé le Recuit Simulé qui est une métaheuristique inspirée par la technique expérimentale de recuit dans la métallurgie. C'est une méthode probabiliste qui permet d'approcher le niveau optimal globale pour une fonction donnée.

Le travail de [27] consiste à sélectionner la meilleure composition des SW à base de l'algorithme de Recherche Harmonique. HS « *Hamony Search Algorithm* » est une métaheuristique basée sur la performance musicale pour trouver l'harmonie parfait, et pour y parvenir, de nombreuses notes différentes sont jouées par les musiciens au sein d'un orchestre

jusqu'à ce que la meilleure harmonie soit obtenue. Les auteurs ont suivi le même principe pour obtenir la meilleure combinaison de SW.

[2] A utilisé dans son travail trois méthodes basées sur la dominance floue : Top-K, Skyline et une méthode hybride FTop-KS composée de Skyline et Top-K pour sélectionner les meilleurs SW composites à partir de plusieurs SW existants. Pour atteindre cet objectif, plusieurs étapes doivent être suivies, en commençant par définir l'ensemble des services disponibles et en les organisant dans des classes, puis appliquant l'un des algorithmes mentionnés ci-dessus afin de choisir le meilleur SW au sein de chaque classe, après avoir obtenu ce dernier, il est ajouté à la liste des SW composites qui sera envoyé à l'utilisateur.

[19] A proposé des méthodes complètement différentes qui ne s'inspirent pas de la nature et des êtres vivants, mais basées sur l'analyse multicritères. Les méthodes AMCD (Analyse Multicritères à la Décision) visent à résoudre des problèmes en considérant un ensemble cohérent de critères pour permettre simultanément la conception, la justification et la transformation des préférences. Celles-ci sont souvent contradictoires et d'importance inégale. —L'auteur a utilisé une méthode de vote appelée Borda pour obtenir une solution de compromis à partir des classements obtenus par les méthodes AMCD, comme chaque méthode AMCD donne un classement différent des SW, vient ensuite le rôle de Borda qui propose un classement final à partir des autres classements.

Ces différentes approches cherchent à optimiser la sélection des SW composite, le tableau 2 montre les méthodes adoptées par chacun des travaux mentionnés ci-dessus ainsi les bases de données utilisées et les résultats déduits des analyses expérimentales sont aussi présentées.

Tableau 2 – Différentes approches relative à la sélection des SW

Référence	Méthode	Type de sélection	Critères	BDD utilisée	Résultats
[25]	NSGA II	Globale	Coût, Temp de réponse, fiabilité, disponibilité, réputation	Fichier XML (10 classes 40 fournisseurs)	Acceptable, temps de réponse non raisonnable si la taille de population est grande.
[22]	Algorithme Mimétique	Hybride	Coût, Temp de réponse, réputation	Fichier XML (120 SW)	Les fonctions de fitness de l'algorithme mimétique sont meilleures par rapport aux valeurs de GA. Un temps d'exécution élevé et taux de croisement élevé. Ne tient pas compte de la fiabilité et de la disponibilité.
[11]	Bees Algorithm	Globale	Coût, Temp de réponse, fiabilité, disponibilité, réputation	(Fichier XML 10 classes 40 fournisseurs)	Résultats relatifs et ne peuvent pas être devinés même s'il garde les mêmes paramètres d'entrées il a obtenu des résultats différents qui peuvent s'améliorer ou diminuer tout dépend du point de départ (population) s'il est mauvais, il ne peut jamais atteindre l'objectif.
[2]	Skyline Top-k FTOP-KS	Hybride	Coût, temp de réponse, fiabilité, débit	Non cité	L'absence des critères Disponibilité et Réputation. Skyline nécessite une amélioration de pour réduire le temps d'exécution.

Référence	Méthode	Type de sélection	Critères	BDD utilisée	Résultats
[15]	FOA	Globale	Coût, temp de réponse, réputation	Fichier XML (10 classes 200 SW)	Résultats acceptables. Ne tient pas compte de la fiabilité et de la disponibilité.
[26]	Recuit simulé	Globale	Coût, temp de réponse, fiabilité, disponibilité, réputation	Fichier XML (10 classes 40 fournisseurs)	Le temps d'exécution diminue et augmente en fonction du nombre d'itérations. L'application ne donne pas toujours de bons résultats.
[27]	Recherche harmonique	Globale	Coût, temp de réponse, fiabilité, disponibilité, réputation	Fichier XML (10 classes 40 fournisseurs)	Un taux d'optimalité mauvais. Les performances chutent de manière exponentielle si le nombre de classes augmente.
[19]	Borda COPRAS SAW VIKOR TOPSIS	Hybride	Coût, temp de réponse, fiabilité, disponibilité, réputation	BDD réelle avec 2507 SW	Résultats acceptables avec un temps d'exécution acceptable.

2.6 Discussion

Le tableau présenté dans la section précédente, résume les différentes méthodes précédemment utilisées dans l'optimisation et la sélection des services Web appropriée pour faciliter la comparaison :

- La colonne Méthodes : fait référence aux méthodes utilisées pour la sélection des SW ;
- Type de sélection : représente la classification (locale, globale ou hybride) de chaque méthode utilisée ;
- La colonne Critères : représente les critères de qualité de service sur lesquels repose la sélection. Le tableau montre que chaque travail soumis comprenait un nombre spécifique de critères. Par exemple, [15] et [22] s'appuyaient sur trois critères le coût, le temps de réponse et la réputation, [2] a s'appuyé sur quatre critères le coût, le temps de réponse, la fiabilité et le début, tandis que les autres s'appuyaient sur cinq critères, ajoutant la disponibilité ;
- La colonne BDD : indique la source des données (base de données) utilisée.

Le tableau représente aussi un résumé sur les résultats obtenus dans chaque méthode, nous remarquons que la majorité des résultats sont acceptables mais avec un certain inconvénients tel que NSGA II et l'algorithme mimétique qui ont donné des résultats meilleurs que GA mais avec un temps d'exécution non raisonnable et un taux de croisement élevé si la taille de la population est grande ce qui nous montre que l'GA nécessite une amélioration ou une hybridation avec d'autres méthodes de sélection pour que les résultats soit satisfaisant. Bees Algorithm ou l'algorithme de colonies d'abeilles à la possibilité de ne pas atteindre l'objectif s'il commence avec une mauvaise population ainsi que le recuit simulé qui ne donne pas toujours de bons résultats.

Le problème de la taille de données en entrée, le temps de réponse et le taux d'optimalité existe aussi dans les autres méthodes tel que la recherche harmonique et FTop-SK. Loin des algorithmes de sélection des SW, les méthodes AMCD ont prouvé leurs efficacités de sélection pour trouver les SW qui correspondent aux exigences de l'utilisateur. Ainsi la méthode de vote Borda qui a donné un meilleur classement dans un temps très raisonnable.

A partir de cette étude, nous avons choisi l'algorithme génétique comme une approche de sélection des SW, car il a prouvé son efficacité sur un large domaine d'application, surtout s'il est combiné avec d'autres méthodes de recherche. En plus de cela, un l'algorithme génétique ne nécessite pas beaucoup de connaissances mathématiques sur le problème à

optimiser, il utilise seulement l'évaluation de la fonction objective ce qui le mettre facile à mettre en œuvre.

La méthode de vote Borda a prouvé aussi son efficacité et rapidité pour le classement, pour cela, nous proposons une approche hybride, cette approche combine entre la méthode de vote de Scrutin de Condorcet dans la phase de sélection locale et l'algorithme génétique dans la phase de sélection globale

2.7 Conclusion

Dans ce chapitre, Nous avons parlé de la sélection des SW ainsi les différentes stratégies de sélection qui existent, en suite nous avons parlé des problèmes d'amélioration et expliqué les méthodes les plus importantes pour résoudre ces problèmes, à savoir : l'heuristique et la méta-heuristique. Après nous avons présenté une étude sur les différents travaux relatifs à la sélection multi objectif de SW. Cette étude nous a permis de fixer les objectifs et de définir les besoins de notre travail qui sera présenté en détail dans les prochains chapitres.

Chapitre 3

Conception du système

3.1 Introduction

L'objectif de notre projet est de sélectionner les meilleures compositions possibles des Services Web et qui sont capables de répondre aux demandes des utilisateurs. Pour atteindre notre but, nous avons proposé une approche de sélection hybride qui combine une recherche locale basée sur la méthode de vote Condorcet avec un algorithme de recherche global qui est l'algorithme génétique.

Dans ce chapitre, nous allons présenter la description du principe de fonctionnement de notre système commençant par son architecture globale, après nous allons détailler les deux approches utilisées. Ensuite, nous allons décrire les différentes étapes de la conception.

3.2 Architecture du système

Cette partie représente l'architecture de notre approche proposée pour la sélection des SW composites à partir de trois services web, les composants de notre système sont :

- 1- Un service qui fournit les informations nécessaires pour la réservation d'un hôtel ;
- 2- Un service qui fournit les informations pour la location de voiture ;
- 3- Un service pour la participation à un loisir ;
- 4- La méthode de vote Condorcet ;
- 5- L'algorithme génétique ;
- 6- Bases de données hôtel, voiture et loisir ;
- 7- Un serveur cloud « Clever cloud ».

La figure 11 présente l'architecture de notre :

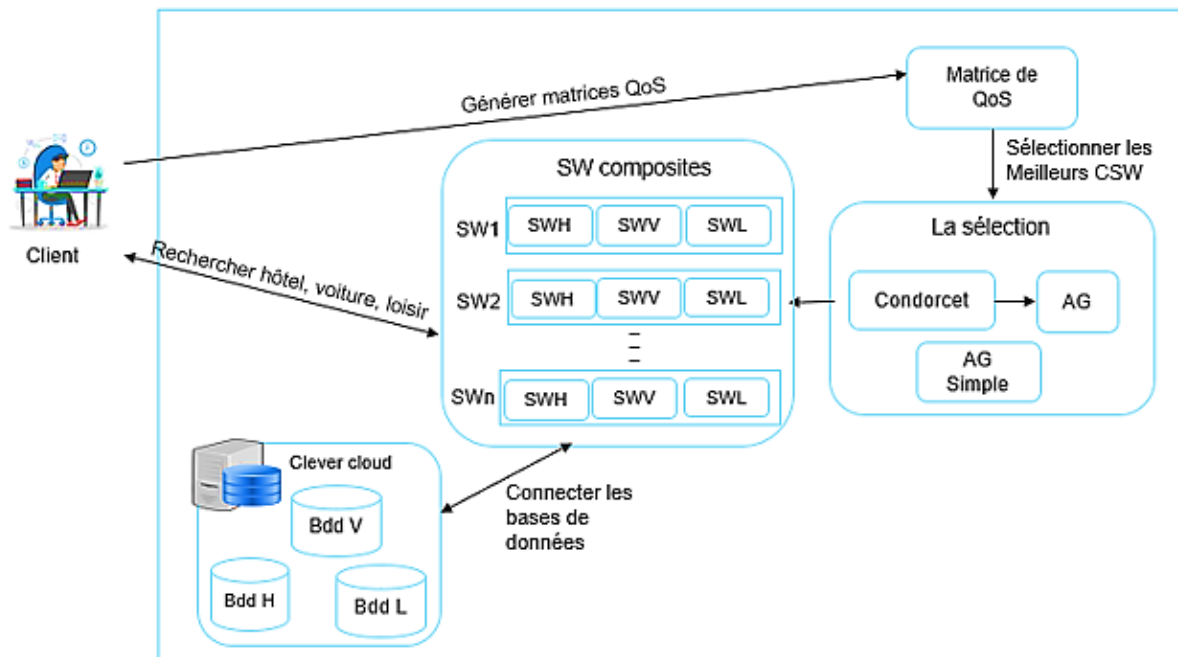


Figure 11 – Architecture du système.

3.3 Organigramme du système

Le système doit choisir les meilleures compositions de services web sur la base des critères de qualité de service, comme montre la figure 12 le processus de notre système est le suivant :

- Générer les matrices des critères de QoS ;
- La sélection se fait à l'aide de l'approche proposée, elle commence par le vote de Condorcet après on applique un algorithme génétique basé sur les contraintes d'utilisateur. Les résultats obtenus seront comparés par la suite avec l'application seulement de l'algorithme génétique ;
- Les meilleurs SW composites seront utilisés pour répondre à la demande de client qui cherche à trouver un hôtel, une voiture et un loisir selon la date de disponibilité, wilaya, et le prix ;
- Les bases de données qui contient les informations nécessaires pour répondre à cette requête se trouve sur un serveur cloud (Clever cloud) ;
- Une fois les résultats de la recherche trouvés, ils ont envoyé au client.

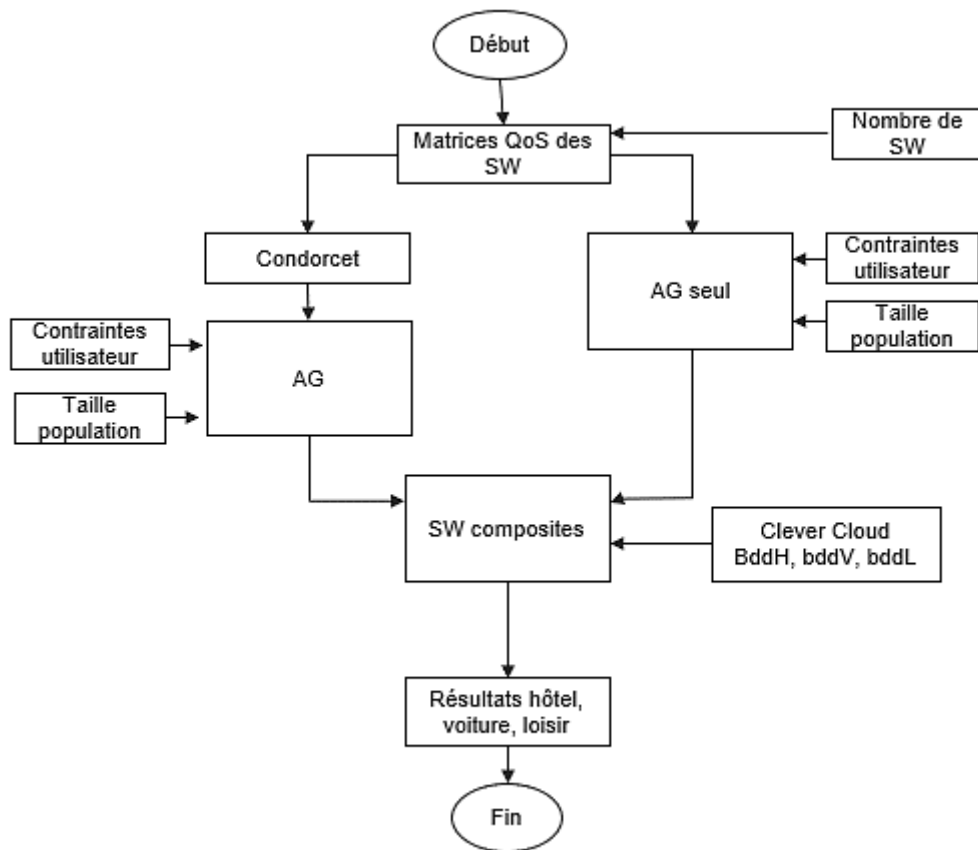


Figure 12 – Organigramme du système.

3.4 Condorcet

Le Scrutin de Condorcet ou vote Condorcet est une technique de vote dans laquelle chaque électeur dresse une liste des candidats par ordre de préférence. Le principe de Condorcet est que « si un candidat est préféré à tous les autres par une majorité, alors il doit être élu » [50] [28].

3.4.1 Fonctionnement de Condorcet

Nous devons classer toute la liste des SW disponible selon un ordre décroissant en commençant par le service le plus préféré jusqu'à le moins préféré. Pour atteindre cet objectif nous avons ajouté une étape de classement à la fin de cette approche.

Supposant que la liste des candidats est la liste des SW disponibles et les électeurs sont les cinq critères de QoS cités précédemment. Cette technique de recherche locale sera appliquée à chaque liste des SW (hôtel, voiture, et loisir) séparément comme le montre le tableau 3 :

Etape 1 : consiste à déterminer la liste des SW candidats.

Tableau 3 – Exemple matrice de candidats

Nom SW	Coût	Latence	Disponibilité	Fiabilité	Réputation
SW 1	4.9773895	57.1891343	2.2849879	0.4534232	0.4771148
SW 2	21.8936897	99.2740406	2.9251300	0.8048730	0.7414196
SW 3	1.1874412	97.7684752	3.7412944	0.4641673	0.2201043
SW 4	15.4259759	272.4760437	4.4429357	0.8638187	0.7760180

Etape 2 : Déterminer la matrice de vote à partir de la matrice des SW candidats, on classe les SW par ordre de préférence des Critères (nous allons minimiser le coût, la latence et maximiser la fiabilité, la disponibilité et la réputation) et remplacer chaque valeur de critère par leur SW correspondant (tableau 4).

Tableau 4 – Exemple matrice de vote

Préférence	Coût	Latence	Disponibilité	Fiabilité	Réputation
Choix 1	SW 3	SW 1	SW 4	SW 4	SW 4
Choix 2	SW 1	SW 3	SW 3	SW 2	SW 2
Choix 3	SW 4	SW 2	SW 2	SW 3	SW 1
Choix 4	SW 2	SW 4	SW 1	SW 1	SW 3

Etape 3 : Calculer de la matrice de choix, cette matrice représente les résultats de la préférence entre chaque paire de SW tandis que le SW le plus haut est le meilleure que l'autre et gagne un point (tableau 5).

Tableau 5 – Exemple matrice de choix

Candidats	SW 1	SW 2	SW 3	SW 4	Somme des points
SW 1	4	2	2	2	11
SW 2	3	4	2	1	11
SW 3	3	3	4	2	13
SW 4	3	4	3	4	15

Etape 4 : En fonction des résultats de la matrice de choix, nous allons classer les SW du plus préféré (qui obtient le maximum de points) au moins préféré (tableau 6). Les résultats de classement de vote de Condorcet sont considérés comme entrée de l’algorithme génétique

Tableau 6 – Exemple résultat Condorcet.

Classement	Coût	Latence	Disponibilité	Fiabilité	Réputation
SW 4	15.4259759	272.4760437	4.4429357	0.8638187	0.7760180
SW 3	1.1874412	97.7684752	3.7412944	0.4641673	0.2201043
SW 1	4.9773895	57.1891343	2.2849879	0.4534232	0.4771148
SW 2	21.8936897	99.2740406	2.9251300	0.8048730	0.7414196

3.4.2 Avantages et inconvénients de Scrutin de Condorcet

Le scrutin de Condorcet permet de trouver une solution de compromis qui satisfait une grande portion des électeurs mais cette méthode de vote n'est pas toujours opérationnelle, dans certains cas, il n'existe pas de gagnant (le paradoxe de Condorcet) par exemple : le candidat (A) est préféré au candidat (B). Ce dernier est préféré au candidat (C), mais la confrontation de (A) et de (C) révèle que (C) est préféré à (A) ce qui nécessite l'utilisation d'autre méthode pour résoudre le problème [29] .

3.5 Algorithme Génétique

AG est un algorithme évolutionnaire itératif de recherche globale qui combine une stratégie de « suivre le plus fort » avec un échange d'informations aléatoire mais organisé. Il est utilisé pour résoudre des problèmes dont la solution est inconnue, où un ensemble de solutions aléatoires est généré appelé une population et chaque solution est associée à un individu, ce dernier est évalué et classé par l'application des différentes opérations génétiques (la sélection, le croisement et la mutation) dont le but est de converger vers la solution optimale [30] [31].

3.5.1 Fonctionnement d'un algorithme génétique

Un AG est défini par ces éléments principaux comme l'illustre la figure 13 :

- Une population (ou génération) : est constituée d'un groupe d'individus (solutions) où chaque individu a un seul chromosome,
- Un chromosome correspond à une chaîne de caractère appelé gène et qui se caractérise par une position sur le chromosome et une valeur du gène.

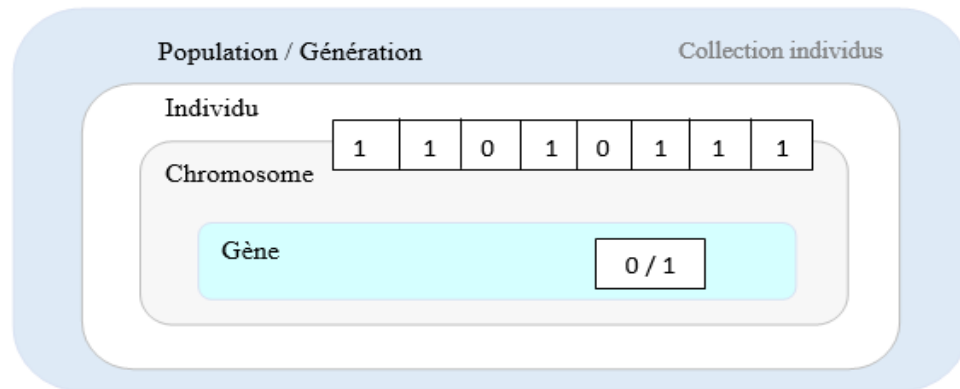


Figure 13 – Eléments principaux dans d'un algorithme génétique [32]

Le principe de fonctionnement de l'AG s'appuie sur plusieurs étapes principales comme illustre la figure 14 :

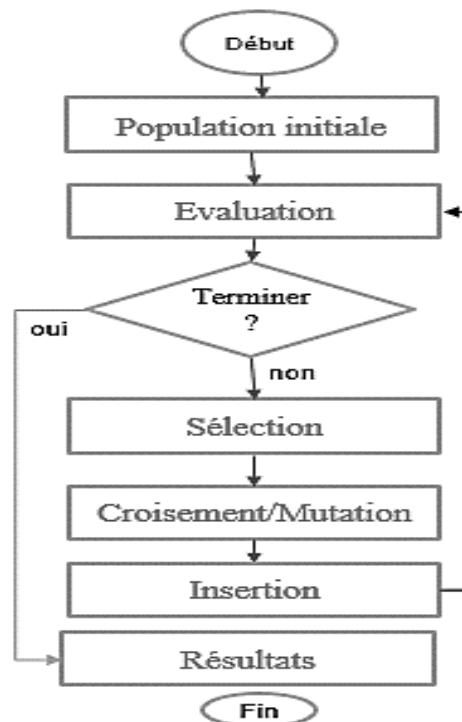


Figure 14 – Organigramme d'Algorithme génétique

- 1- Définition de la population initiale P qui génère un nombre N de solutions aléatoires (individus) pour un problème donné.
- 2- Evaluation de chaque individu est réalisée par une fonction objective qui détermine la capacité d'un individu à rivaliser avec d'autres individus.
- 3- Opérateur de sélection définit l'ensemble des individus de la population P qui vont être dupliqués dans la nouvelle population P' et les considérés comme des parents afin d'appliquer les autres opérateurs génétiques croisement et mutation.
- 4- Puis une transaction est créée entre les solutions parentes et les solutions produites par un mécanisme d'insertion [33].

3.5.1.1 Codage des données

C'est une modélisation de la résolution d'un problème particulier sous la forme d'une série de chaîne de caractères appelée chromosome qui est un ensemble de gènes. Il consiste à sélectionner le contenu génétique (contenus des gènes) qui facilite la description du problème et respecte ses contraintes. Il existe trois types de codage, le codage numérique, le codage symbolique et le codage alphanumérique. Nous focalisons sur le codage numérique qui est divisé en deux types de codage principales [33] [34] :

Codage binaire : dans un codage binaire chaque chromosome est représenté par une chaîne de bit ou chaque élément prend la valeur 0 ou 1.

Codage réel : dans ce type de codage, le gène est représenté par une suite de bit qui est associé à un réel. Donc un chromosome est représenté comme suit :

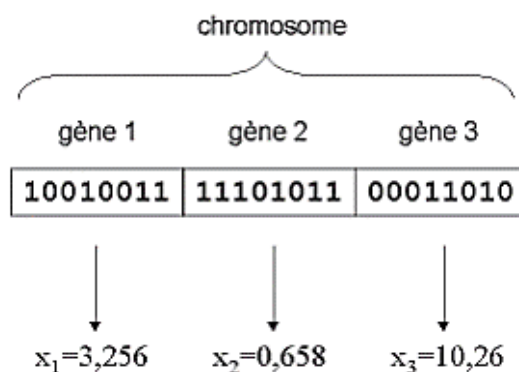


Figure 15 – Exemple du codage réel d'un chromosome [35]

Le codage réel est utilisé généralement dans le cas où l'on recherche le maximum (ou le minimum selon le problème à optimiser) d'une fonction réelle.

Dans notre approche, nous supposons que le chromosome (individu) est composé de plusieurs SW ou chaque service comporte cinq critères de QoS. Chaque gène représente la valeur d'un critère et pour cela nous avons besoin d'un codage réel qui va nous permettre de maximiser une fonction réelle.

3.5.1.2 Initialisation de la population

La population initiale représente l'ensemble des solutions possibles et l'entrée d'un AG. La sélection de la taille et de la bonne population contribue à la rapidité, l'optimalité et à la convergence de l'algorithme [34]. A cause de cette importance nous avons combiné l'AG avec la méthode de recherche locale Condorcet.

Cette méthode de vote nous aidera à améliorer la génération de la population initiale, elle consiste à classer l'ensembles des SW existants (hôtels, voitures et loisir) selon les différents critères de QoS en éliminant les mauvais services avant de définir la population initiale. Comme nous avons vus dans la section 3.4 Condorcet donne un classement local pour chaque SW ce que nécessite la combinaison des trois services, la figure 16 explique le processus de génération de la population initiale :

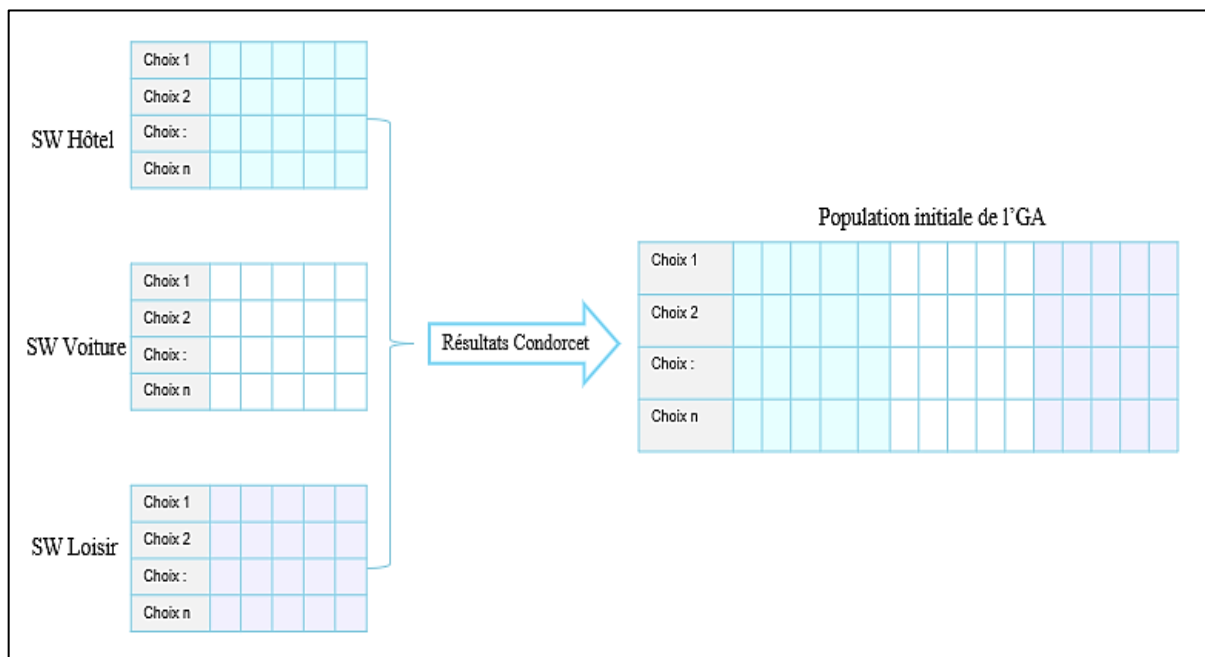


Figure 16 – Génération de la population initiale

3.5.1.3 Evaluation (fitness)

Après la création de la population initiale chaque individu doit être évalué selon une fonction d'évaluation ou fitness, cette fonction décrit la performance des individus et elle permet de les comparer afin de déterminer quels individus seront sélectionnés pour être reproduits ou s'ils doivent être remplacés [36].

Chaque individu est composé de trois services ou chaque service possède des critères de QoS. Nous focalisons dans notre système sur les critères suivants : le coût, la latence, la fiabilité, la disponibilité et la réputation, et pour évaluer chaque individu nous avons besoin de calculer la qualité de chaque SW composé (individu) en commençant par la définition des fonctions d'agrégation pour chaque critère (tableau 7).

Tableau 7 – Les fonctions d'agrégation des critères de QoS [37]

Critères	Fonctions d'agrégation
Coût	$cout(sol) = \sum_{i=1}^n coût(ci) \quad (1)$
Temp de réponse (latence)	$Latence(sol) = \sum_{i=1}^n lat(ci) \quad (2)$
Fiabilité	$Fiab(sol) = \log_2 \prod_{i=1}^n fiab(ci) \quad (3)$
Disponibilité	$Disp(sol) = \log_2 \prod_{i=1}^n dis(ci) \quad (4)$
Réputation	$Rép(sol) = \frac{1}{n} \sum_{i=1}^n rép(ci) \quad (5)$

Ces critères doivent optimiser une fonction globale qui minimise le coût et latence et maximise la fiabilité, disponibilité et la réputation et nous devons également satisfaire des contraintes globales comme suit :

- Coût (sol) < coût définit par l'utilisateur ;
- Latence (sol) < latence définit par l'utilisateur ;
- Fiabilité (sol) > Fiabilité définit par l'utilisateur ;
- Disponibilité (sol) > Disponibilité définit par l'utilisateur ;
- Réputation (sol) > réputation définit par l'utilisateur.

Selon [25] la qualité de SW composé et la fonction d'évaluation peuvent être décrites comme suit :

$$QoS(sol) = 1/f(cout(sol), lat(sol), fiab(sol), dis(sol), rep(sol)) \quad (6)$$

$$f(\text{sol}) = \frac{1}{|P*\text{cout}(\text{sol})-P*\text{lat}(\text{sol})-P*\text{fiab}(\text{sol})-P*\text{dis}(\text{sol})-P*\text{rep}(\text{sol})|} \quad (7)$$

P : le poids de chaque critère de QoS.

3.5.1.4 Sélection

L'opérateur de sélection ou reproduction identifie les individus de P les plus forts et susceptibles d'obtenir les meilleurs résultats et qui seront répliqués dans la nouvelle génération p' et utilisés comme des parents pour la reproduction. Cette notion est généralement liée avec la valeur de la fonction d'évaluation car chaque individu est sélectionné en fonction de sa valeur de fitness. Donc, les individus qui présentent une performance élevée, ont d'autant plus de chance d'être sélectionnés, tandis que ceux qui sont moins bien adaptés seront éliminés [38].

Il existe de nombreuses techniques de sélection [35] :

a. Sélection élitiste (classement)

Elle consiste à sélectionner les individus nécessaires pour la nouvelle génération P' en prenant les plus forts individus de la population P après les avoir triés de manière décroissante (ou croissante selon l'objectif).

b. Sélection par tournoi

Cette méthode consiste à choisir aléatoirement deux individus ou plus puis leurs fonctions d'adaptation sont comparées et la mieux adaptée est sélectionnée. Cette procédure est répétée jusqu'à ce que la nouvelle population soit complète.

c. Sélection uniforme

Elle consiste à choisir aléatoirement les individus sans faire intervenir leurs valeurs d'adaptations. Donc chaque individu dans la population P a une probabilité d'être sélectionné, mais la convergence de l'algorithme sera lente [34].

d. Sélection par roulette

C'est la méthode la plus populaire pour les sélections aléatoires elle consiste à associer à chaque individu un segment dont la longueur est proportionnelle à son fitness, ces segments sont ensuite concaténés sur un disque (roue de loterie).

Dans notre système, la sélection de l'ensemble des individus de la population P qui vont être dupliqués dans la nouvelle génération P' se fait par la stratégie de sélection élitiste ou nous allons classer les services d'une manière décroissante.

3.5.1.5 Croisement

C'est le principal opérateur de l'AG qui permet l'échange des informations génétiques entre les individus. Il se produit avec deux parents choisis aléatoirement, de tel sorte que chacun produit deux descendants qui héritent des caractéristiques de leurs parents. Après le croisement, les parents peuvent être retirés de la population de reproduction ou conservés pour une autre chance de se reproduire [33]. Parmi les méthodes de croisement qui existent citons :

- **Le croisement à un point** : consiste à sélectionner aléatoirement un emplacement de locus (point de croisement) et de permuter les parties droites des deux parents comme montre la figure 17.



Figure 17 – Croisement à un point [34]

- **Le croisement multi point** : consiste à sélectionner deux ou plusieurs points de croisement et permuter les différentes parties entre les parents pour former les fils. Figure 18.



Figure 18 – Croisement à multi points [34]

Dans notre système, l'application de l'opérateur de croisement permet de créer des nouvelles solutions possibles par l'échange des services composites entre deux individus (figure 19), et nous allons appliquer un croisement à un point en raison de la taille assez réduite du chromosome, ce point va être choisi aléatoirement à chaque opération de croisement ce qui permet l'échange d'un service parmi les services qui composent l'individu.

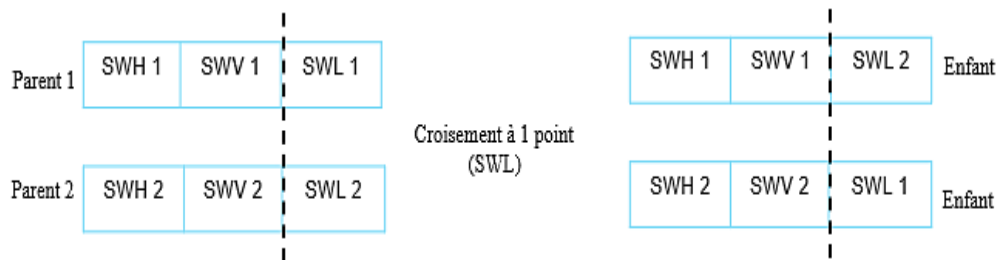


Figure 19 – Exemple de croisement dans notre système

3.5.1.6 Mutation

Dans un algorithme génétique, la modification d'une partie de la population par l'opérateur de mutation permet l'exploration de tout l'espace de recherche. Elle permet de garder une diversité dans l'évolution des individus et d'éviter les optimums locaux.

La mutation consiste à changer la valeur d'un gène (figure 20) avec une très faible probabilité comprise généralement entre 0.01 et 0.001, cet opérateur contribue à modifier de manière totalement aléatoires les caractéristiques d'une solution ce qui permet d'introduire et de maintenir une diversité au sein de population de solutions [35].

Dans notre système l'opérateur de mutation modifie les valeurs des critères de QoS d'un service parmi les services qui compose une solution.

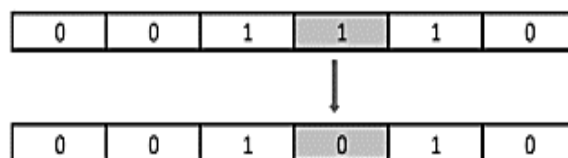


Figure 20 – Opérateur mutation [35]

3.5.1.7 Remplacement

L'opérateur de remplacement réintroduit les descendants dans P' qu'ils ont obtenus par l'applications des trois opérations précédents (sélection, croisement et mutation) dans la population de leurs parents P. c'est l'opérateur le plus simple, son objectif est de remplacer les mauvais individus par des autres pour passer à la nouvelle génération [35].

3.5.2 Paramètres de dimensionnement d'un AG

Un AG nécessite les paramètres internes suivants [38] :

- La taille de la population : c'est le nombre d'individus au cours de chaque génération ;
- Le choix de la taille de la population joue un rôle très important pour la convergence de l'algorithme et l'évaluation de l'individu ;
- Le critère d'arrêt qui indique l'arrêt d'un processus génétique ;
- Les probabilités des opérateurs génétique qui détermine le choix de taux de croisement et de mutation ;
- Le taux de croisement est généralement compris entre 70% et 95% de la population totale, tandis que le taux de mutation compris entre 0.1% et 1% de la population totale.

Ces valeurs sont issues de la littérature ou sur l'expérience acquise progressivement.

3.5.3 Avantages d'algorithme génétique

- Peut être utilisé pour résoudre plusieurs types de problèmes dans différents domaines d'applications tel que l'optimisation dans les réseaux et dans les voyages de commerce, constitution des équipes de travail, la mise en point d'un emploi du temps, etc ;
- Utiliser seulement l'évaluation de la fonction objective, ce ne que nécessite pas beaucoup de connaissances mathématiques car elle est basée sur les techniques d'évolution naturelle ;
- Offre une flexibilité d'hybridation avec d'autre méthodes, ce qui permet de trouver des bons résultats et dans un temps raisonnable ;
- Offre une recherche globale et il travaille sur plusieurs points en même temp (ensemble d'individu) au lieu d'un seul, donc il est efficace si la fonction possède plusieurs optimums locaux [34].

3.5.4 Inconvénients d’algorithme génétique

- Les algorithmes génétiques trouvent des solutions raisonnables et pas nécessairement optimales ;
- Ils sont basés sur une recherche aléatoire ce qui permet de perdre des solutions ;
- Ils ne garantissent pas d’obtenir une solution dans un temps fini ;
- Les opérateurs génétiques de base sont parfois insuffisants pour résoudre un problème Ce qui peut conduire à un mauvais résultat [38].

3.6 Conception

Dans cette section nous allons présenter les différents diagrammes qui donnent la vue de conception de notre système commençons par le diagramme de cas d’utilisation, après les diagrammes de classes et en fin le diagramme de séquence.

3.6.1 Diagramme de cas d’utilisation

La figure 21 représente le diagramme de cas d’utilisation de notre application, qui structure les besoins d’utilisateur et les objectifs de notre système.

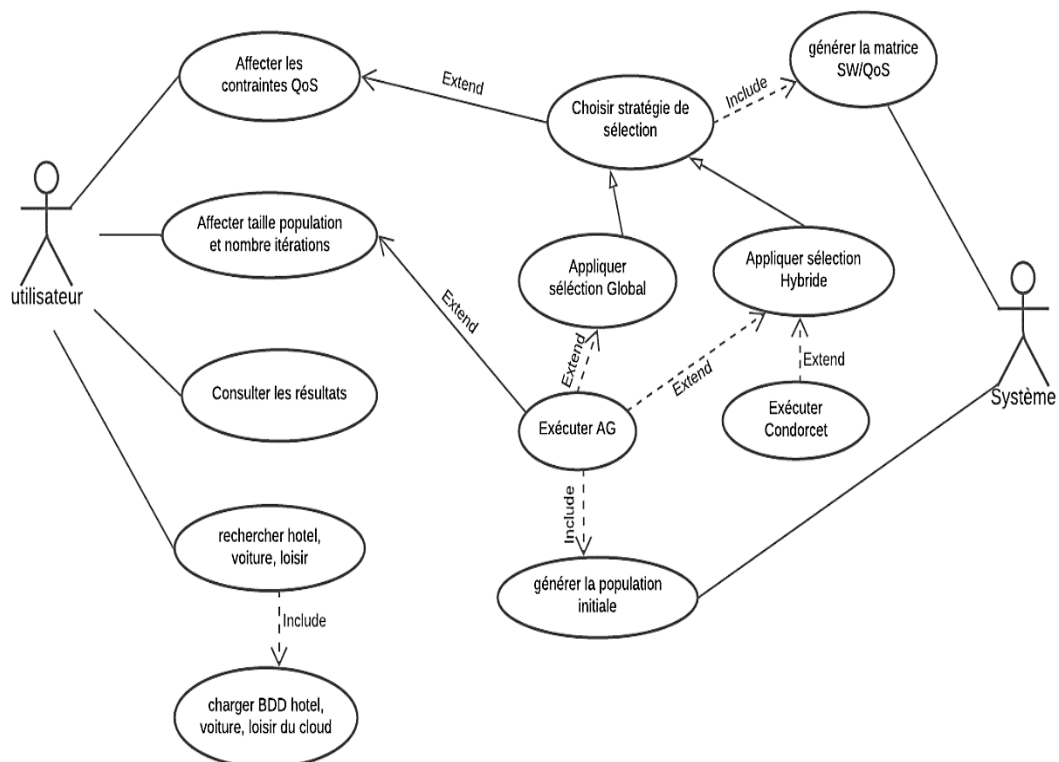


Figure 21 – Diagramme de cas d'utilisation

3.6.2 Diagrammes de classes

Les diagrammes de classes sont utilisés pour représenter les classes et interfaces des systèmes et leurs relations, et dans ce qui suit, nous allons présenter les divers diagrammes de classes liés à notre système. Les figures 22, 23, 24 montrent les diagrammes de classes de nôtres bases de données :

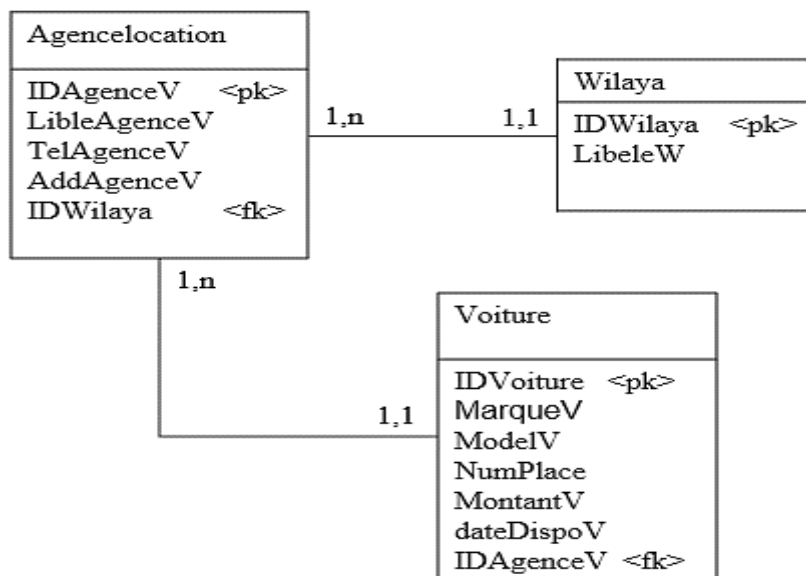


Figure 22 – Diagramme de classe pour la BDD Voiture

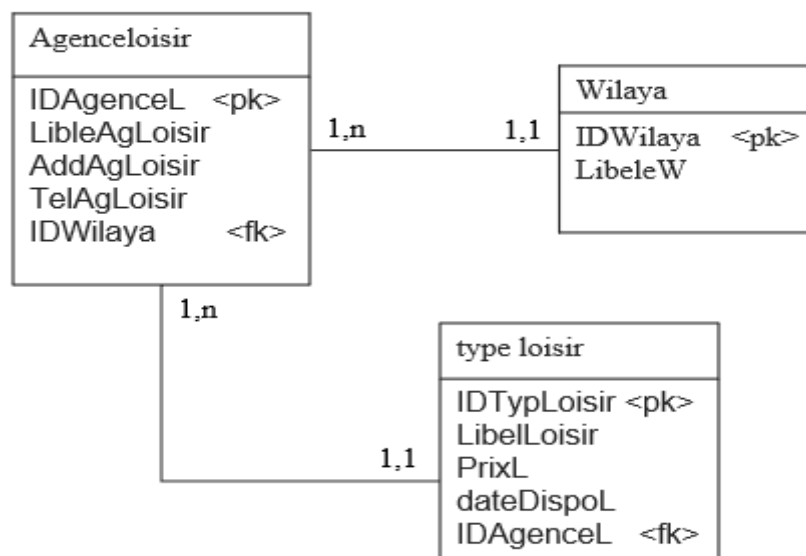


Figure 23 – Diagramme de classes pour la BDD loisir.

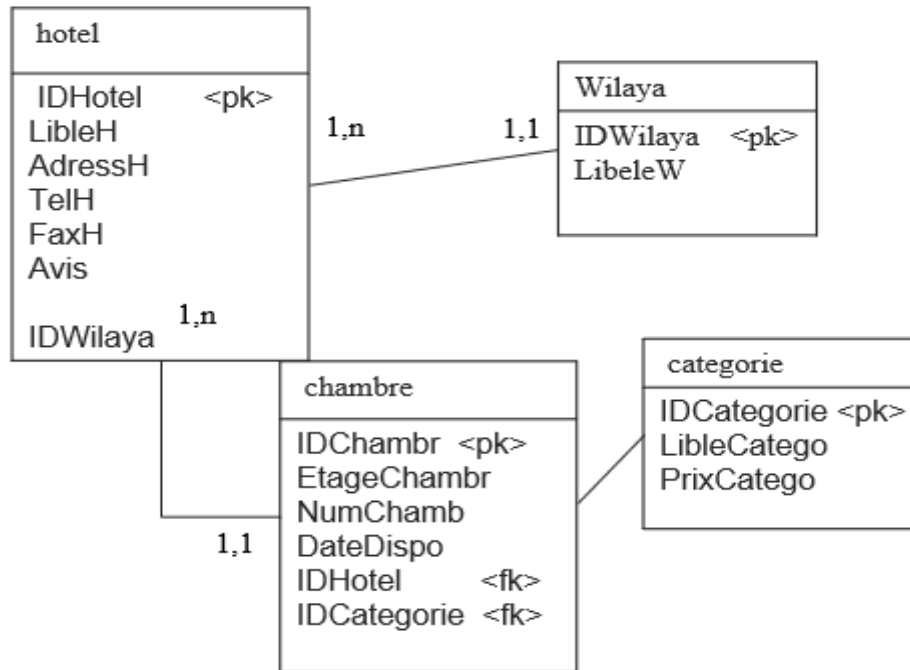


Figure 24 – Diagramme de classe pour la BDD hôtel

3.6.2.1 Diagramme de classe de l'application

Le diagramme (figure 25) représente les classes intervenant dans notre système pour la sélection du SW composite :

- La Classe Condorcet qui représente la méthode de vote Condorcet.
- Une classe principale AlgoGénitique, elle contient un constructeur, la méthode run () qui lance l'algorithme et la méthode finaliser () qui teste les critères d'arrêts (trouver une solution ou le nombre d'itérations atteint).
- La classe Population qui définit la population initiale de notre AG, elle contient deux méthodes :
 - resultat_Condorcet qui définit la population initiale à partir des résultats de Condorcet ;
 - initial_resultat qui définit la population initiale de l'AG sans Condorcet ;

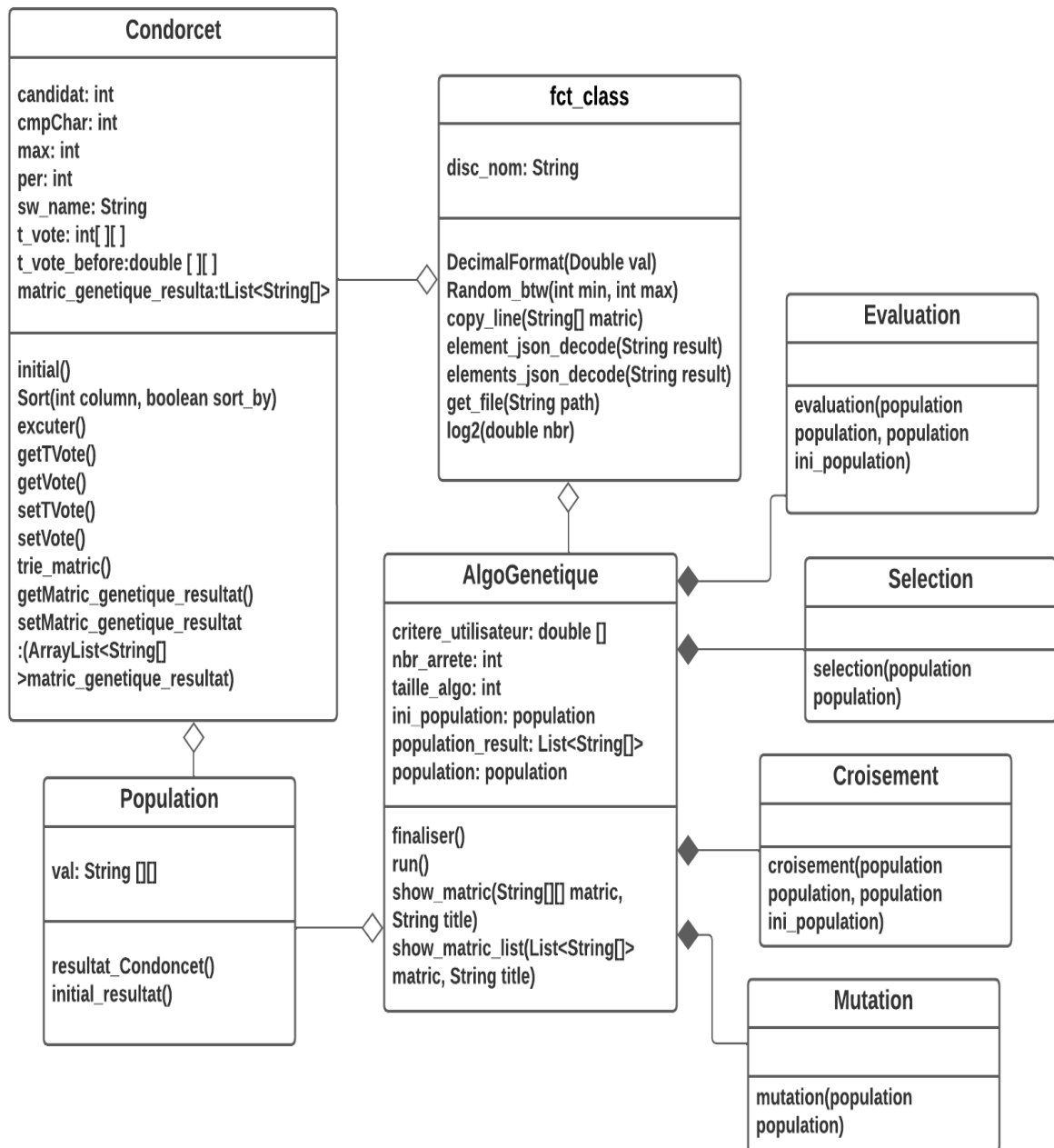


Figure 25 – Diagramme de classes de l'approche proposée

- Les classes des opérations génétiques :

- Evaluation : cette classe permet de calculer les fonctions d'agrégations et le fitness de chaque individu selon la fonction de fitness (voir les pages 33 et 34) ;
- Sélection : cette classe permet de sélectionner la liste des individus qui vont être reproduite par les autres opérateurs génétiques croisement et mutation ;

- Croisement : cette classe consiste à choisir deux parents aléatoirement et appliquer un échange d'un seul SW parmi les trois services qui composent l'individu afin de produire des descendants qui héritent les caractéristiques de leurs parents ;
 - Mutation : cette classe modifier les valeurs des critères d'un seul SW (SWH, SWV ou SWL) aléatoirement.
- La classe fct_class contient les fonctions nécessaires pour faciliter le fonctionnement de notre AG tel que, la fonction mathématique $\log_2(\text{double nbr})$ et la fonction `random_btw (int min, int max)`.

Pseudocode de l'approche proposée :

Input : Nombre SW, critères, taille population, nombre itération

Output : services web composites

1) initialiser les critères des SW `t_vote_before [][]`

2) initialiser la matrice de vote `t_vote [][]`

If (`critère==0`|| `critère==1`)

Trier `t_vote_before [][]` par ordre croissant

Else

Trier `t_vote_before [][]` par ordre décroissant

3) Calculer matrice de choix

4) retourner résultat de Condorcet

5) Initialiser la population initiale `init_population` pour chaque algorithme

a) Population AG avec condorcet `resultat_condorcet()`

b) Population AG seul `resultat_initial()`

6) Evaluer `init_population`

Pour tous les individus

6.2.1.1 Calculer les fonctions d'agrégation

$$Cost(sol) = \sum_{i=1}^n coût(ci)$$

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

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

$$Disp(sol) = \log_2 \prod_{i=1}^n dis(ci)$$

$$Rép(sol) = \frac{1}{n} \sum_{i=1}^n rép(ci)$$

6.2.1.2 Calculer la qualité de service

$$QoS(sol) = 1/f(cout(sol), lat(sol), fiab(sol), dis(sol), rep(sol))$$

6.2.1.3 Calculer le fitness

$$f(sol) = 1/(P*rep(sol) - P*lat(sol) - P*fiab(sol) - P*dis(sol) - P*rep(sol))$$

Fin pour

7) **Tant que** critère d'arrêt non atteint

val_population = sélectionner les parents dans init_population par une sélection élitiste

val_population = appliquer opérateur de croisement à un point sur val_population

val_population = appliquer opérateur de mutation sur val_population

init_population = remplacer les anciens de init_population par leurs descendants de val_population

Fin Tant que

3.6.3 Diagramme de séquence

Ce diagramme représente les interactions entre le système et les acteurs selon un ordre chronologique. La figure 26 montre le diagramme de séquence de notre système

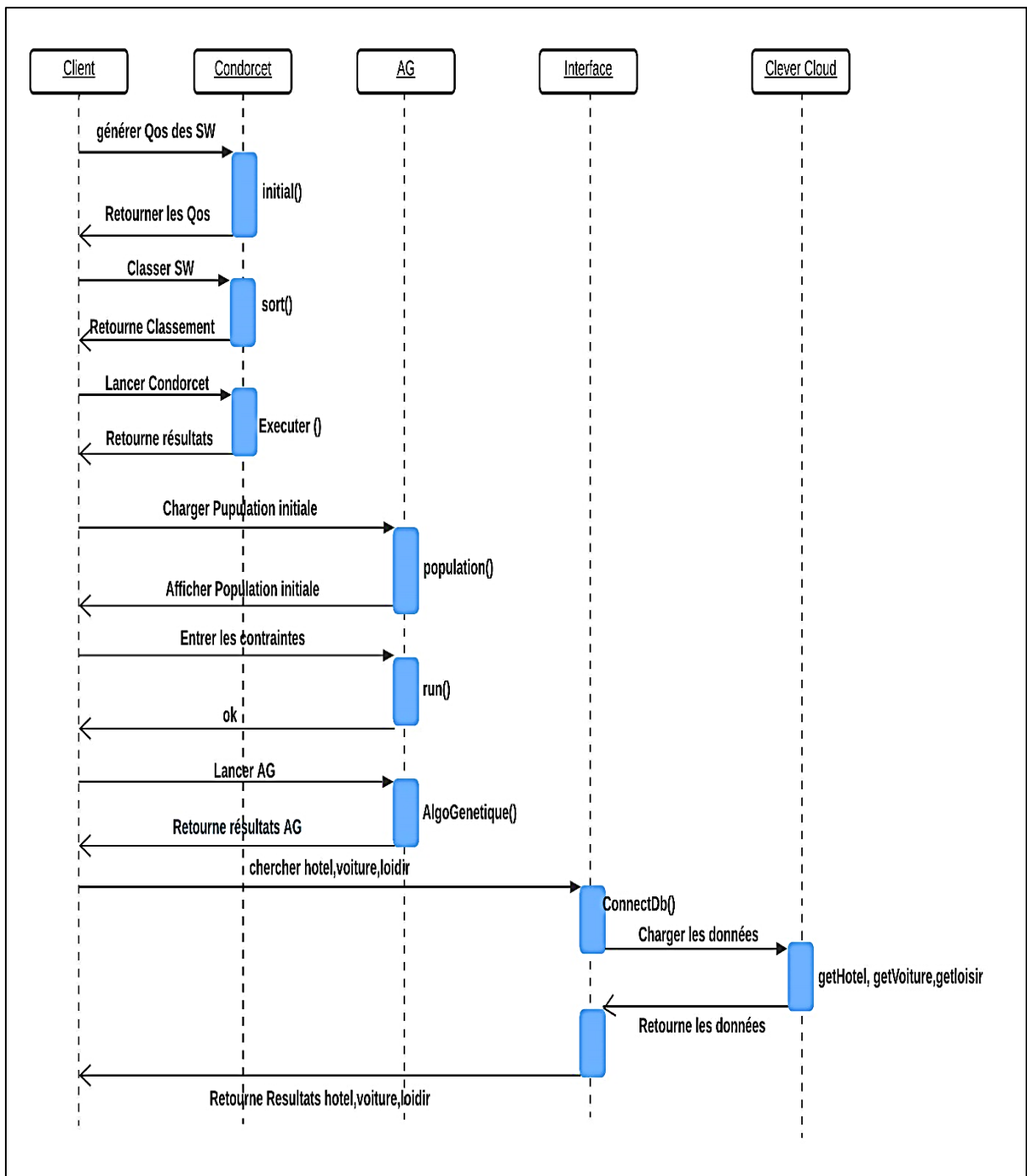


Figure 26 – Diagramme de séquence de l'approche proposée

3.7 Conclusion

Dans ce chapitre nous avons présenté le scénario de notre projet qui est basé sur la sélection des services web ainsi que l'architecture du système et les détails de notre approche ont été expliqués au début du chapitre, après nous avons présenté les différentes étapes de la conception. Le but n'est pas de choisir le meilleur SW mais de trouver un ensemble de

services web composite les plus pratiques et qui satisfaisant les contraintes globales de l'utilisateur. Pour atteindre notre objectif, nous avons appliqué une approche de sélection hybride composée de la méthode de vote Condorcet et d'un algorithme génétique, et nous avons exécuté aussi l'algorithme génétique seul afin de comparer les résultats des deux algorithmes.

Dans le chapitre suivant, nous présentons l'implémentation de notre application et les résultats de l'expérimentation.

Chapitre 4

Implémentation et expérimentations

4.1 Introduction

Ce chapitre représente l'aspect d'implémentation de notre application web et les résultats expérimentaux. Nous allons commencer par la représentation de l'environnement de développement, les langages de programmation et la description de notre base. Ensuite nous décrirons les interfaces et analyserons les résultats finaux.

4.2 Environnement de développement

4.2.1 Environnement Matériel

Notre système a été développé sur un ordinateur portable Fujitsu LIFEBOOK A357 avec les caractéristiques suivants :

- Un processeur Intel(R) Core (TM) i5-7200U CPU @ 2.50GHz 2.70 GHz
- Mémoire RAM 8,00 Go
- Un disque dur SSD 256 Go
- Un système d'exploitation Microsoft Windows 10 Professionnel 64 bits.

4.2.2 Environnement Logiciel

NetBeans

Pour la réalisation de notre application web nous avons choisi l'environnement de développement intégré NetBeans pour



java qui permet le développement et le déploiement rapide et gratuits des applications. Notre choix de NetBeans dû à notre expérience dans le développement d'applications java [39].

Clever Cloud

Il s'agit d'un service d'hébergements de nouvelles générations qui gère et sécurise les environnements de développements. Clever Cloud prend en charge un large éventail d'environnements d'exécution, de bases de données et de fonctionnalités, disponibles à la fois sur un cloud public et sur notre propre cloud privé.



Nous avons utilisé cette plateforme PaaS (Platform as a Service) pour la création et la gestion de nos bases de données hotel_db, voiture_db et loisir_db qui contiennent les informations nécessaires pour la recherche d'un hôtel, voiture et loisir. Clever Cloud nous a beaucoup aidé grâce à sa facilité d'utilisation et à la mise à disposition de l'application web PhpMyAdmin de gestion pour les systèmes de gestion de bases de données relationnelles MySQL [40].

Gson

Google Search Organization Network, est une bibliothèque Java Open Source développé par Google, utilisée pour analyser et générer les fichier JSON. Elle permet de convertir des objets Java en leur représentation JSON et vice versa [41].



```
1 package newpackage;
2
3 import com.google.gson.Gson;
   import com.google.gson.reflect.TypeToken;
```

Figure 27 – Code d'importation de la bibliothèque Gson

JDBC

Java DataBase Connectivity désigne une API (Application Programming Interface) pour permettre un accès aux bases de données SQL (Structured Query Language) avec Java. Elle établit la connexion avec une base de données relationnelle come MySQL, envoie les requêtes SQL au serveur et traite les résultats retournés par l'exécution de la requête [42].



JSP

Java Server Pages est une technologie développée par Sun Microsystems en 1999, cette technologie est spatialement utilisée pour aider les développeurs de créer des pages Web générées de manière dynamique (telles que HTML et XML) [43].



Notre choix de technologie JSP dû à notre expérience et il peut connecter à toute type de bases de données avec JDBC.

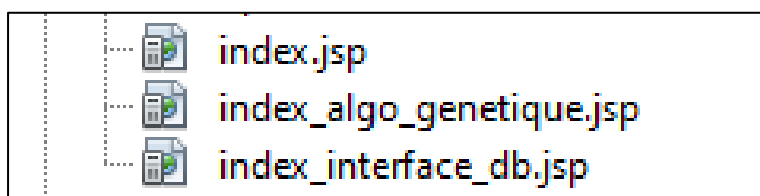


Figure 28 – Pages JSP de notre Application Web

4.2.3 Langages de programmations

4.2.3.1 Java

Est un langage de programmation orienté objet et plateforme informatique créés par Sun Microsystems en 1995 [44]. Ce langage permet le lancement et l'exécution de plusieurs taches (threads) simultanément dans les applications. Cela garantie une utilisation efficace des ressources matérielles, réduire le temps de réponse, le cout et donne une meilleure performance [45].



Notre choix de langage Java dû à notre expérience dans le développement d'applications, il est simple, sécurisé et permet de créer des projets standard et du code réutilisable, il offre aussi la possibilité d'effectuer plusieurs taches simultanément (Multithread).

4.2.3.2 JavaScript

Est un langage de programmation de script orienté objet créé en 1995 par Brendan Eich utilisé pour implémenter des mécanismes complexes sur les pages web. Il sert à ajouter de l'interactivité entre la page web et l'utilisateur ou navigateur, à l'aide de JavaScript nous pouvons : gérer des formulaires d'une manière plus avancé que celle proposé en HTML, développer des animations complexes, créer des textes, des contenus défilants, des dessins, des jeux, etc. [46].



Avantages : JavaScript a la capacité de prendre en charge tous les navigateurs modernes et à produire un résultat équivalent, il fonctionne bien avec d'autres langages et peut être utilisé dans un grand nombre d'applications. Il économise beaucoup de bande passante et accélère le processus d'exécution [47].

Inconvénients : JavaScript peut être difficile pour développer de grandes applications, son code est toujours visible par tout le monde, et si une erreur se produit dans le JavaScript, il peut s'arrêter pour rendre l'ensemble du site Web [47].

Notre choix de langage JavaScript dû à notre expérience dans le développement d'applications, ainsi JavaScript donne le pouvoir de faire des interfaces riches et il est utilisé partout sur le web.

4.2.3.3 HTML

HyperText Markup Language est le langage universel utilisé pour communiquer sur le Web, il est utilisé pour créer et représenter le contenu d'une page web et sa structure [48].



Nous avons utilisé le langage HTML car il est facile à mettre en œuvre, permet de gérer des informations diverses et dû à notre expérience dans le développement d'applications.

4.2.3.4 JSON

Est un format de données indépendant de tout langage utilisé pour l'échange et la représentation des données textuelles et structurées [49].



Avantage : Ce format texte est facile à lire, à écrire et analyser grâce à une syntaxe simple et à une structure en arborescence. Un fichier JSON permet aussi de stocker les informations et les transférer entre un serveur et un client. JSON est exploitable directement avec JavaScript.

Inconvénients : JSON ne permet pas de présenter tous les types de données comme les dates et les couleurs et il ne permet pas les commentaires.

Nous avons utilisé JSON parce qu'il est simple à mettre en œuvre, la vitesse du traitement et nous permet de stocker les données de types nombres, tableau (Array), etc.

4.2.4 Description de la BDD

Notre matrice des SW/QoS est un fichier JSON. Chaque service est caractérisé par cinq critères de QoS et les données du fichier sont générées aléatoirement de telle manière que le coût est pris dans l'intervalle [0-30], la latence entre [0-300], la fiabilité entre [0-5], la disponibilité et la réputation entre [0-1].

L'objectif de ce travail est de trouver un ensemble de service web composites qui minimise le coût, la latence et maximise la fiabilité, disponibilité et la réputation, les borne de ces critères sont définit par l'utilisateur.

```
Gson gsonObj = new Gson();
H_map_value = gsonObj.toJson(H_map);
V_map_value = gsonObj.toJson(V_map);
L_map_value = gsonObj.toJson(L_map);

is_exists(fct_class.disk_nom + "H_Random.json", gsonObj.toJson(H_map_value).replace("\"", ""));
is_exists(fct_class.disk_nom + "V_Random.json", gsonObj.toJson(V_map_value).replace("\"", ""));
is_exists(fct_class.disk_nom + "L_Random.json", gsonObj.toJson(L_map_value).replace("\"", ""));
```

Figure 29 – Code source pour créer nos fichiers JSON

Nous avons utilisé aussi Clever Cloud pour la création et la gestion de nos bases de données Hôtel, Voiture et Loisir (Figure 30) qui contiennent les informations essentielles pour la recherche du client.

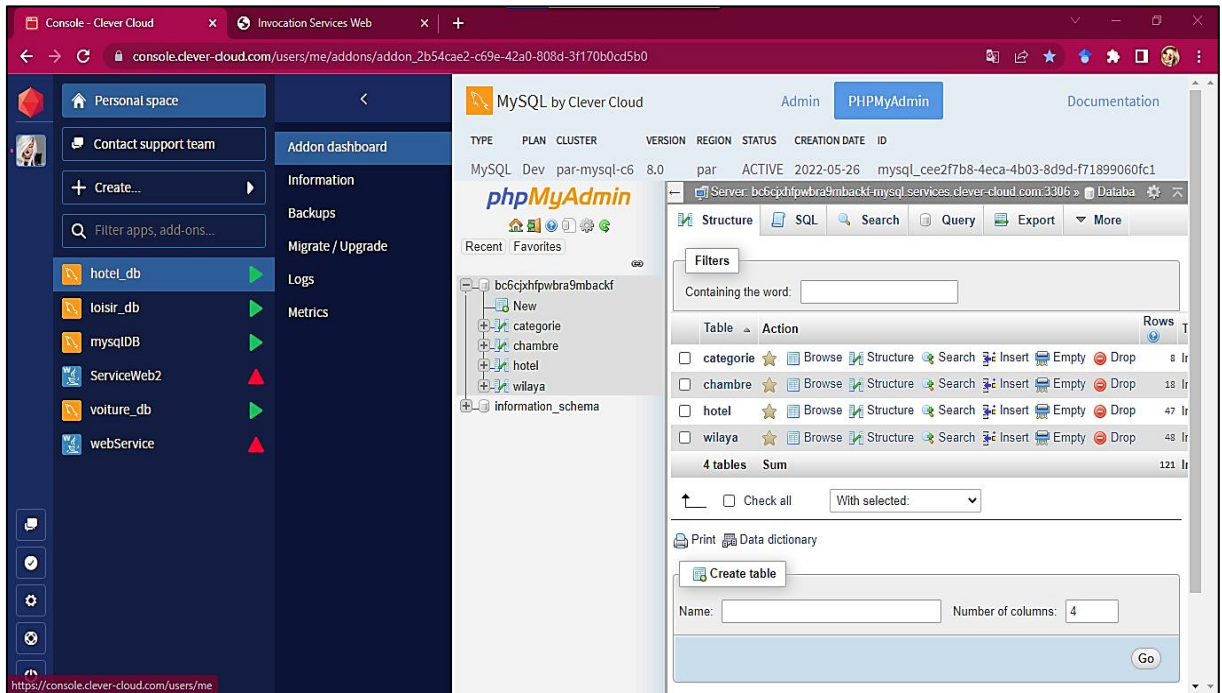


Figure 30 – Interface Clever cloud pour hotel_db

4.3 Présentation des interfaces

Dans cette section, nous allons donner quelques captures d'écrans de notre application web, commençons par la première fenêtre qui représente la méthode de vote de Condorcet. Cette fenêtre est divisée en deux parties :

- **Charger les matrices de SW/QoS** : Dans cette partie l'utilisateur doit préciser le nombre de services web nécessaire. Le bouton « Charger » nous permet de charger les matrices des critères des services web pour chaque classe hôtel, voiture et loisir (Figure 31).

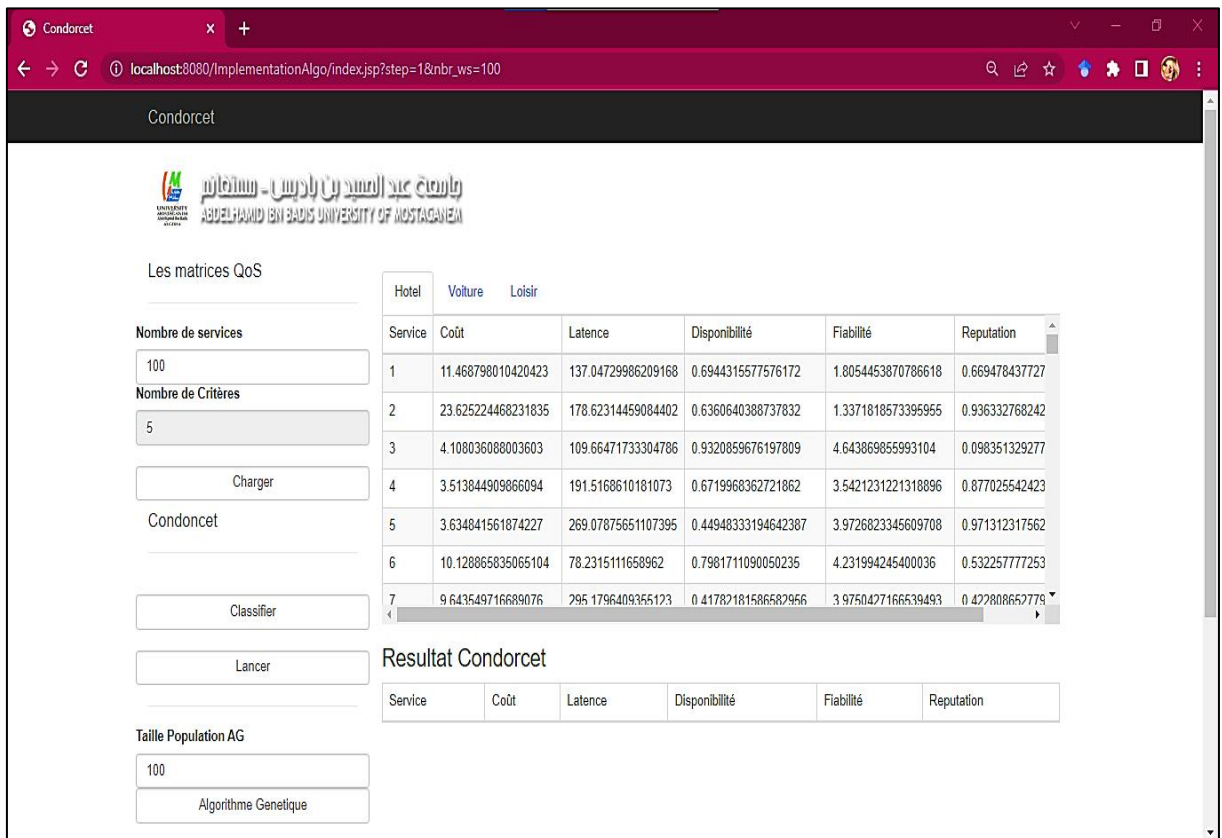


Figure 31 – Chargement des matrices de QoS

Le bouton « Classifier » permet de classer les SW de chaque classe selon l'ordre de préférence pour chaque critère de QoS (Figure 32).

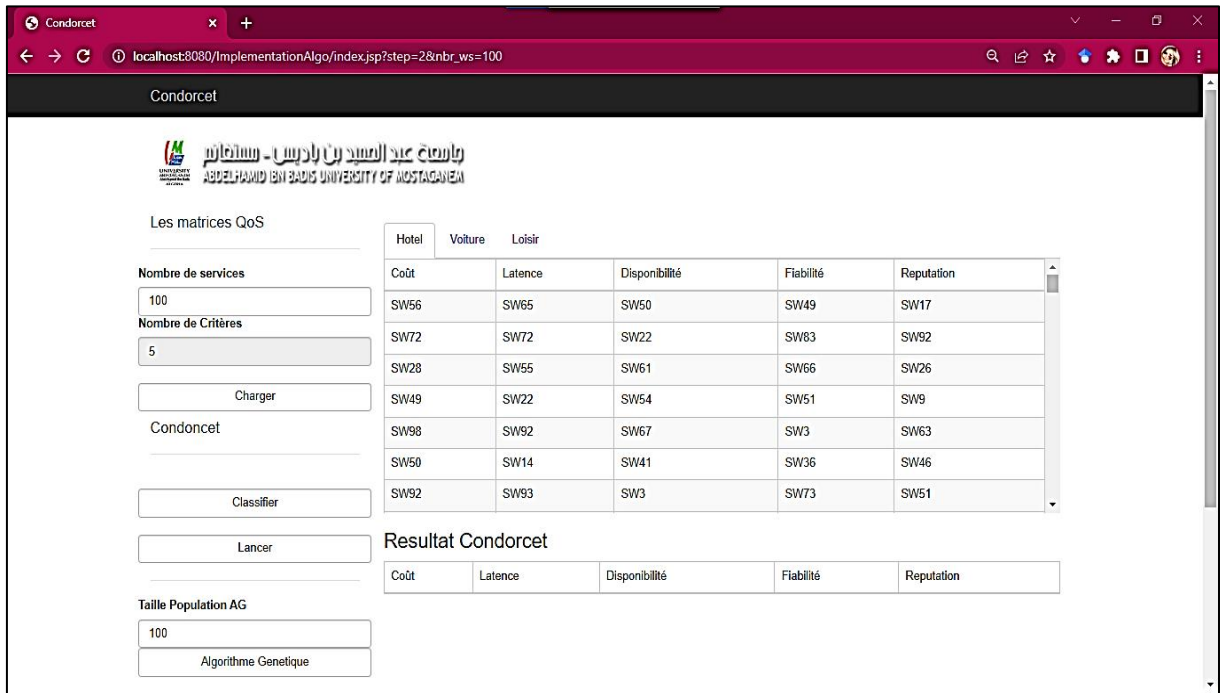


Figure 32 – Classement de SW par ordre de préférence

- **Méthode de vote de Condorcet :** Dans cette partie on lance la méthode Condorcet à l'aide du bouton « lancer ». Une fois le système termine les calculs, il retourne les résultats (Figure 33).

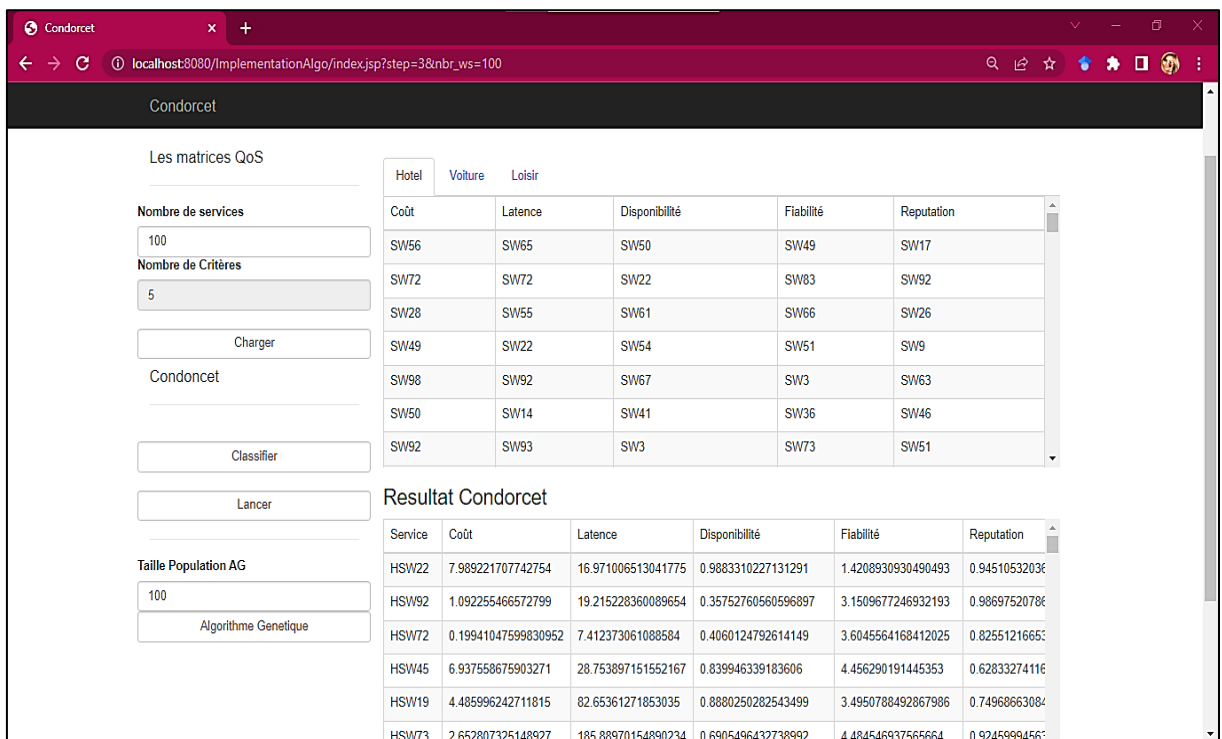


Figure 33 – Résultats de Condorcet

Avant de passer à la deuxième fenêtre, l'utilisateur doit spécifier la taille de la population initiale de l'algorithme génétique puis il clique sur le bouton « algorithme génétique » qui permet le passage à la fenêtre suivante.

La deuxième fenêtre de notre système représente la méthode de recherche globale basé sur l'algorithme génétique, elle est composée de deux parties aussi. La première partie représente notre méthode hybride qui va combiner le vote de Condorcet avec l'AG dans la deuxième partie nous allons appliquer un AG sans le combiner avec la méthode de recherche locale. Le critère de test d'arrêt doit être défini par l'utilisateur, ainsi les valeurs de borne de chaque critère de QoS (Figure 34).

Algorithme Génétique

UNIVERSITY OF MOSTAGANEM
ABDELHAMID IBN BADIS UNIVERSITY OF MOSTAGANEM

Algorithme Génétique

Algorithme genetique + condorcet Algorithme genetique

Service	Coût	Temp d'exécution	Disponibilité	Fiabilité	Reputation
HSW22	7.989221707742754	16.971006513041775	0.9883310227131291	1.4208930930490493	0.94510532036
HSW92	1.092255466572799	19.215228360089654	0.35752760560596897	3.1509677246932193	0.98697520786
HSW72	0.19941047599830952	7.412373061088584	0.4060124792614149	3.6045564168412025	0.82551216653
HSW45	6.937558675903271	28.753897151552167	0.839946339183606	4.456290191445353	0.62833274116
HSW19	4.485996242711815	82.65361271853035	0.8880250282543499	3.4950788492867986	0.74968663084
HSW73	2.652807325148927	185.88970154890234	0.6905496432738992	4.484546937565664	0.92459994563
HSW46	4.031677366298657	117.8167885304292	0.8627796486911203	1.3404165321871686	0.97599489794

Resultat AG + Condorcet (0)

Service	Coût	Temp d'exécution	Disponibilité	Fiabilité	Reputation	Service	Coût	Temp d'exécution	Disponibilité	Fiabilité

Lancer

Lancer service

Figure 34 – Population initiale de l'AG avec Condorcet

Une fois que l'utilisateur saisi les informations nécessaires pour notre recherche, il suffit de cliquer sur le bouton « lancer » pour lancer notre AG hybride et L'AG seul, les résultats sont affichés immédiatement après que les algorithmes obtiennent des solutions ou le critère d'arrêt est atteint. (Figure 35, 36).

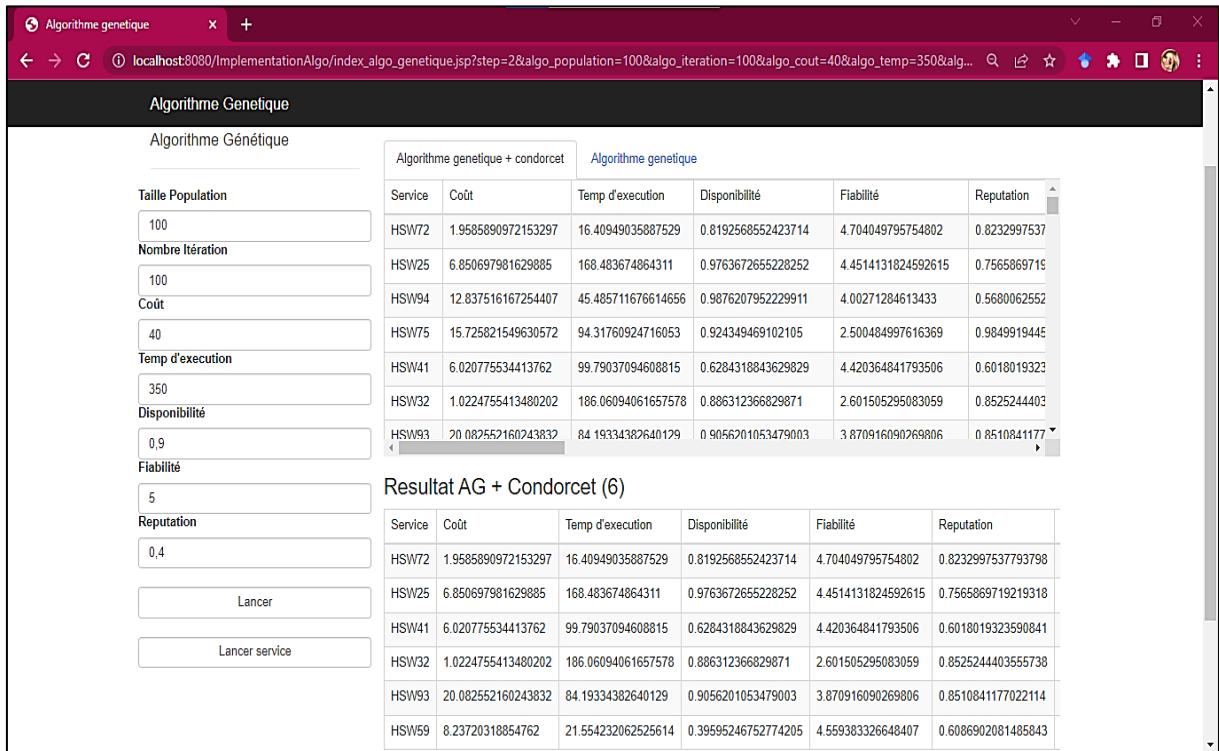


Figure 35 – Résultats AG + Condorcet

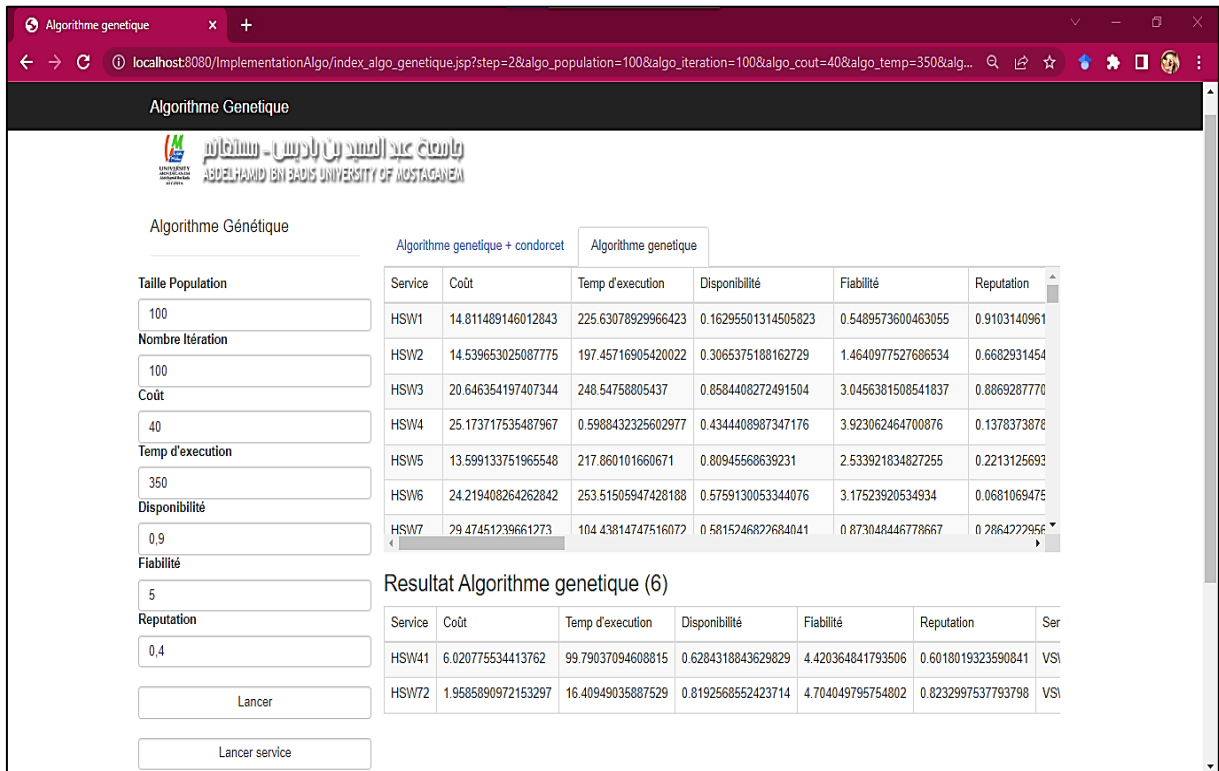


Figure 36 – Résultats AG

Le bouton « lancer service » sur la deuxième fenêtre permet le passage à la troisième fenêtre (figure 37). C’est l’interface principale de notre système qui invoque les services web sélectionnés par les précédents algorithmes afin d’obtenir les résultats de la recherche du client. Elle fournit à l’utilisateur un service de recherche d’un hôtel, voiture ou loisir selon la wilaya, la date de disponibilité et le prix.

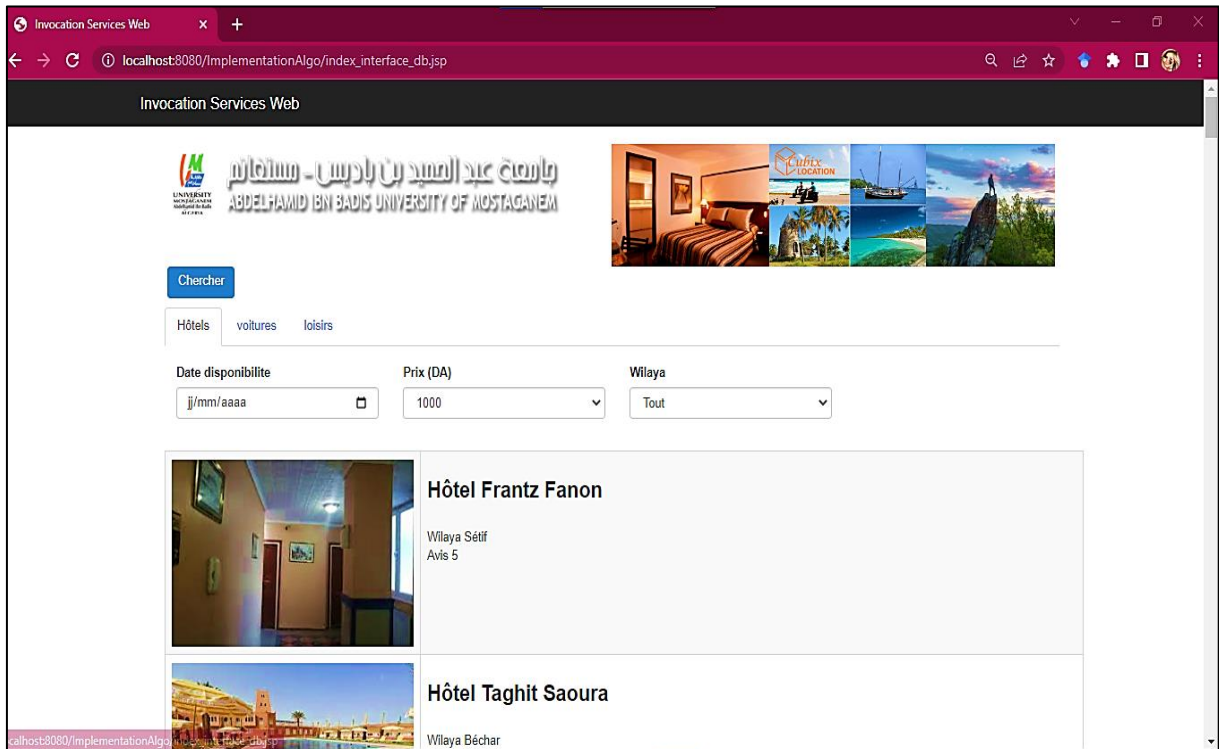


Figure 37 – Invocation des services Web

4.4 Expérimentation

Nous avons mené une expérience pour évaluer la performance de notre approche proposée, cette expérience est une comparaison entre les résultats obtenus par l’AG combiné avec Condorcet et l’AG simple. Les résultats de fitness, le temps d’exécution et le nombre de solution (SW composite) des deux algorithmes sont présentés dans les tableaux 8, 9, 10, 11, 12, 13 :

Fixer :

Taille population = 50

Nombre d’itération = 100

Tableau 8 – Résultats première configuration

	Algorithme Hybride (Condorcet + AG)			Algorithme Génétique		
	Temp d'exécution	Fitness (Max)	Résultats	Temp d'exécution	Fitness (Max)	Résultats
<i>Simulation 1</i>	0.005s	0.021	7 CSW	0.005s	0.004	1 CSW
<i>Simulation 2</i>	0.008s	0.004	1 CSW	0.14s	/	/
<i>Simulation 3</i>	0.010s	0.006	5 CSW	0.007s	0.005	2 CSW
<i>Simulation 4</i>	0.007S	0.012	3 CSW	2.672S	/	/
<i>Simulation 5</i>	0.005s	0.009	4 CSW	0.004s	0.003	1 CSW
<i>Simulation 6</i>	0.005s	0.005	3 CSW	2.52s	/	/
<i>Simulation 7</i>	0.006s	0.005	2 CSW	2.12s	/	/
<i>Simulation 8</i>	0.006s	0.015	6 CSW	2.21s	/	/
<i>Simulation 9</i>	0.008s	0.007	1 CSW	2.35s	/	/
<i>Simulation 10</i>	0.005s	0.009	1 CSW	2.33s	/	/

Dans le tableau 8 on remarque que dans la plupart des simulations l'algorithme génétique seul ne donne pas des résultats contrairement à notre approche qui donne plusieurs résultats et dans un temps raisonnable et meilleur que l'AG seul.

Fixer :

Taille population = 100

Nombre d'itération = 100

Tableau 9 – Résultats deuxième configuration

	Algorithme Hybride (Condorcet + AG)			Algorithme Génétique		
	Temp d'exécution	Fitness (Max)	Résultats	Temp d'exécution	Fitness (Max)	Résultats
<i>Simulation 1</i>	0.009s	0.007	5 CSW	0.013s	0.003	1 CSW
<i>Simulation 2</i>	0.006s	0.020	9 CSW	16.21s	/	/
<i>Simulation 3</i>	0.008s	0.006	2 CSW	18.98s	/	/
<i>Simulation 4</i>	0.009s	0.009	5 CSW	0.008s	0.003	1 CSW

<i>Simulation 5</i>	0.010s	0.009	7 CSW	0.006s	0.005	2 CSW
<i>Simulation 6</i>	0.005s	0.016	4 CSW	0.007s	0.004	1 CSW
<i>Simulation 7</i>	0.009s	0.009	7 CSW	0.005s	0.003	1 CSW
<i>Simulation 8</i>	0.008s	0.005	6 CSW	0.006s	0.007	1 CSW
<i>Simulation 9</i>	0.005s	0.014	9 CSW	0.008s	0.004	2 CSW
<i>Simulation 10</i>	0.008s	0.012	5 CSW	0.007s	0.004	2 CSW

Nous remarquons que dans la deuxième et la troisième simulation l'AG seul ne trouve pas des résultats et les temps d'exécution jusqu'à terminer 100 itérations n'est pas raisonnable. Dans les autres simulations, le temps d'exécution de notre approche est meilleur que de l'AG et parfois l'inverse.

Fixer :

Taille population = 150

Nombre d'itération = 100

Tableau 10 – Résultats de la troisième configuration

	Algorithme Hybride (Condorcet + AG)			Algorithme Génétique		
	Temp d'exécution	Fitness (Max)	Résultats	Temp d'exécution	Fitness (Max)	Résultats
<i>Simulation 1</i>	0.011s	0.010	9 CSW	0.009s	0.011	4 CSW
<i>Simulation 2</i>	0.011s	0.013	10 CSW	68.51s	/	/
<i>Simulation 3</i>	0.010s	0.011	7 CSW	0.009s	0.007	1 CSW
<i>Simulation 4</i>	0.012s	0.013	12 CSW	0.008s	0.003	2 CSW
<i>Simulation 5</i>	0.011s	0.022	7 CSW	0.010s	0.017	2 CSW
<i>Simulation 6</i>	0.011s	0.017	5 CSW	0.009s	0.004	2 CSW
<i>Simulation 7</i>	0.010s	0.013	9 CSW	0.009s	0.005	1 CSW
<i>Simulation 8</i>	0.010s	0.013	10 CSW	0.014s	0.003	1 CSW
<i>Simulation 9</i>	0.010s	0.018	11 CSW	0.009s	0.004	2 CSW
<i>Simulation 10</i>	0.011s	0.020	7 CSW	0.015s	0.006	2 CSW

Fixer :

Taille population = 200

Nombre d'itération = 100

Tableau 11 – Résultats de la quatrième configuration

	Algorithme Hybride (Condorcet + AG)			Algorithme Génétique		
	Temp d'exécution	Fitness (Max)	Résultats	Temp d'exécution	Fitness (Max)	Résultats
<i>Simulation 1</i>	0.014s	0.011	14 CSW	0.015s	0.006	2 CSW
<i>Simulation 2</i>	0.012s	0.018	8 CSW	0.008s	0.006	2 CSW
<i>Simulation 3</i>	0.011s	0.009	11 CSW	0.019s	0.003	1 CSW
<i>Simulation 4</i>	0.030s	0.014	7 CSW	0.031s	0.003	2 CSW
<i>Simulation 5</i>	0.015s	0.009	8 CSW	0.012s	0.005	2 CSW
<i>Simulation 6</i>	0.021s	0.012	13 CSW	0.016s	0.009	1 CSW
<i>Simulation 7</i>	0.012s	0.005	10 CSW	0.020s	0.004	1 CSW
<i>Simulation 8</i>	0.014s	0.011	10 CSW	0.018s	0.007	2 CSW
<i>Simulation 9</i>	0.016s	0.020	14 CSW	0.012s	0.008	1 CSW
<i>Simulation 10</i>	0.012s	0.023	9 CSW	0.012s	0.003	1 CSW

Fixer :

Taille population = 250

Nombre d'itération = 100

Tableau 12 – Résultats de la cinquième configuration

	Algorithme Hybride (Condorcet + AG)			Algorithme Génétique		
	Temp d'exécution	Fitness (Max)	Résultats	Temp d'exécution	Fitness (Max)	Résultats
<i>Simulation 1</i>	0.018s	0.011	17 CSW	0.013s	0.008	2 CSW
<i>Simulation 2</i>	0.016s	0.031	9 CSW	0.019s	0.004	2 CSW

<i>Simulation 3</i>	0.016s	0.010	10 CSW	0.013s	0.004	6 CSW
<i>Simulation 4</i>	0.016s	0.010	11 CSW	0.019s	0.013	5 CSW
<i>Simulation 5</i>	0.018s	0.021	11 CSW	0.016s	0.004	3 CSW
<i>Simulation 6</i>	0.015s	0.017	8 CSW	0.015s	0.005	4 CSW
<i>Simulation 7</i>	0.014s	0.011	12 CSW	0.014s	0.005	3 CSW
<i>Simulation 8</i>	0.018s	0.047	13 CSW	0.014s	0.004	3 CSW
<i>Simulation 9</i>	0.016s	0.009	14 CSW	0.014s	0.008	4 CSW
<i>Simulation 10</i>	0.016ss	0.010	8 CSW	0.014s	0.003	2 CSW

Le tableau 13 représente les valeurs maximales de fitness et de temps d'exécution obtenues pour chaque méthode de sélection en fonction de la taille de la population afin de faciliter la comparaison entre les deux algorithmes génétique simple et combiné avec Condorcet.

Tableau 13 – Représentation des maximums pour chaque population

	AG - Condorcet		AG	
	Temps d'exécution (Max)	Fitness (Max)	Temps d'exécution (Max)	Fitness (Max)
<i>Population 50</i>	0.006s	0.015	0.007s	0.005
<i>Population 100</i>	0.006s	0.020	0.006s	0.007
<i>Population 150</i>	0.011s	0.022	0.010s	0.017
<i>Population 200</i>	0.012s	0.023	0.016s	0.009
<i>Population 250</i>	0.018s	0.047	0.013s	0.008

Les résultats obtenus sont présentés dans les histogrammes (figure 38, 39).

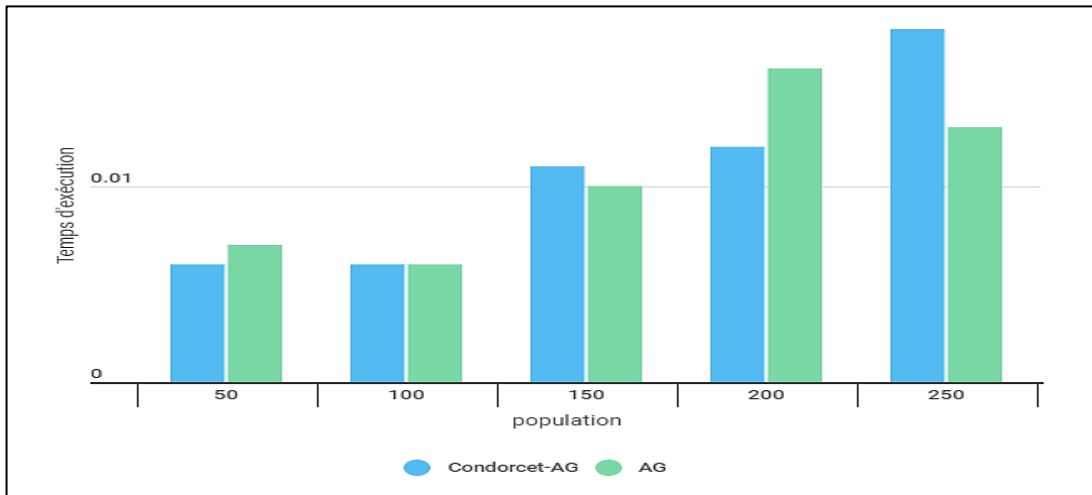


Figure 38 – Histogramme de comparaison de temps d'exécution

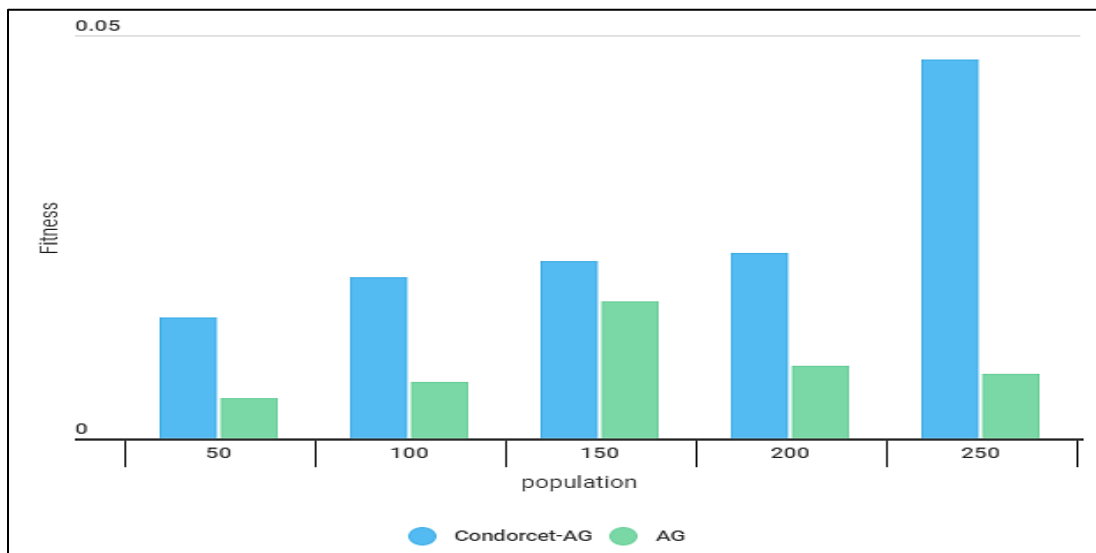


Figure 39 – Histogramme de comparaison de fitness

Discussions :

Nous avons vu que :

- Les valeurs de la fonction fitness de l'approche proposée sont meilleurs par rapport aux valeurs de l'algorithme génétique seul.
- Notre approche proposée donne plus de résultats que l'algorithme génétique seul. On remarque dans certain cas, l'AG n'a pas trouvé de solution (tableau 8)

- Selon l'histogramme présenté (figure 38), le temps d'exécution varie d'une population à une autre et nous pouvons également conclure qu'il est lié aux valeurs de critères de qualité de services (le contenu des individus) et pas seulement la taille de la population.

En comparant les résultats obtenus, nous pouvons conclure que les résultats sont acceptables voir satisfaisant par rapport à l'exécution de l'algorithme génétique sans classement préalable.

4.5 Conclusion

Ce dernier chapitre a été consacré à la présentation de notre application web, commençons par la description des environnement matériels et logiciels ainsi les langages de programmations et la description de nos bases de données. Les interfaces de notre application sont implémentées d'une façon simple, compréhensibles et facile à utiliser dont le but de bien décrire les étapes de la sélection des meilleurs services web composites à l'aide de notre approche proposée et qui est composé de la méthode de vote Condorcet et l'algorithme génétique. A la fin de ce chapitre, nous avons présenté les résultats obtenus par la méthode hybride (Condorcet et l'AG) et un algorithme génétique afin de montrer l'efficacité de notre approche.

Conclusion Générale

Depuis l'existence du réseau le web est devenu une source d'informations dynamique dans lequel l'information est produite sur demande. Pour cela, les services web sont devenus le facteur le plus attirant pour les entreprises actuelles. Un service web est une technique qui permet à deux ou plusieurs machines (client/serveur) de communiquer entre eux à distance via le réseau dont l'objectif est de répondre aux besoins du client.

Un service web assure l'exécution d'une seule fonctionnalité ce qui le rend incapable de répondre aux exigences du client ce qui nécessite la présence de plusieurs services qui fonctionnent de manière unifiée afin de trouver les résultats souhaités et c'est ce qu'on appelle un service web composite. Le choix de la meilleure composition de service web est lié à plusieurs critères de qualité de services, dans ce manuscrit nous avons présenté au début une étude sur les services web et les méthodes utilisées pour la sélection. Cette étude nous a permis de fixer les objectifs et de définir les besoins de notre travail qui consiste à choisir et retourner les meilleures compositions de services web pour la réservation d'hébergements hôtel, voiture et loisir.

Dans ce travail nous avons proposé une méthode de sélection hybride qui combine une recherche locale basée sur le vote de Condorcet avec un algorithme de recherche global (Algorithme génétique). La recherche est basée sur cinq critères de qualité de service le coût, le temps d'exécution, la fiabilité, la disponibilité et la réputation. Afin d'évaluer notre approche proposée, nous avons aussi utilisé un algorithme génétique sans combinaison avec le vote de Condorcet ou nous avons comparé les résultats obtenus par les deux algorithmes.

Perspectives

Comme perspectives, nous envisageons un travail futur qui concerne les aspects suivants :

- Améliorer le temps d'exécution de notre approche proposée.
- Appliquer d'autres types d'opérateurs génétiques : croisement à deux points, à multi points, sélection par tournoi, etc.

- Faire une sélection basée sur d'autres critères de QoS tel que : Débit, sécurité, scalabilité, etc.
- Implémentation d'autres algorithmes d'optimisation.
- Profiter des fonctionnalités et des avantages de cloud et définir la sélection dans le contexte du Cloud Computing.

Bibliographies

Livres et Monographies

- [1] Taleb, Assia. «Thème Conception et Réalisation pour la médecine de travail basé sur: Les Services Web Sémantique.» Mémoire , Université Abou Bakr Belkaid– Tlemcen, 2012.
- [2] DERRADJI, Hazar. «Une Approche Effective Pour La Sélection Des Services Web Composites.» Mémoire, Université de Abou Bakr Belkaid – Tlemcen, 2019.
- [6] AMARAL Rodrigues, et DIEGO João. Architecture et mise en place de services web dans un environnement sécurisé garantissant la confidentialité des informations d'un centre de contrôle de qualité externe. Memoire, Genève: Haute École de Gestion de Genève (HEG-GE), 2016.
- [10] WOOLLAMS, Kwame. «Les services web.UDDI.» Université Paris-Est Marne la Vallée., 2004.
- [11] BOUROUIS, Meriem, et sabah DJERRIRI . «La sélection de services Web à base de l’algorithme d’abeilles.» Mémoire, Faculté des Sciences - Tidjani Haddam Département d’Informatique., 2013.
- [12] TEMGLIT, Nacéra. «La composition dynamique des services web.» Mémoire, université des sciences et de la technologie Houari Boumediene- Alger., 2019.
- [13] GHARBI, Ghazi. «Algorithme de sélection dans les applications à services : une approche basée sur la méthode d’analyse de concepts formels.» Mémoire, Université joseph fourier, 2010.
- [15] BERRABAH, Samir. «Sélection de service web basée sur IFOA (fruit Fly optimization algorithm).» Mémoire, Université Abou Bekr Belkaid– Tlemcen., 2017.
- [16] EL BOUHISSI BRAHMI, Houda. «Web service.» cours, Informatique, Univercité de bedjaya, Bedjaya, 2021.
- [18] MAHBOOBEH, Moghaddam, et Joseph G DAVIS. «Service Selection in Web Service Composition: A Comparative Review of Existing Approaches.» Dans Web Services Foundations, de A., Sheng, Q., Daniel, F Bouguettaya, 321–346. New York, NY: Springer, 2014.
- [19] SERRAI, Walid Salah Eddine. Évaluation de performances de solutions pour la découverte et la composition des services web. Thèse , Université Paris-Est; Université des Sciences et de la Technologie Houari Boumediene (Algérie)., 2020.
- [21] Hakim, Boudjelaba. Sélection des Web Services Sémantiques . Mémoire, (Doctoral dissertation, univ-bouira. dz; Université Bouira; univ bouira)., 2012.
- [22] HIMA, Faiçal, et Hamza GAID. Sélection de Service Web à base d'algorithme Mimétique. Mémoire, Univercité Echahid Hamma Lakhder dd’El Oued, 2016.

- [23] DERKAOUI, Orkia. Metodes d'optimisaion, l'heuristique et le méta heuristique. Cours, Informatique, Saida: Univercité Docteur Moulay TAHAR de Saida, 2021.
- [24] LEMOUARI, ALI. Introduction aux Métaheuristiques. Mémoire, Faculté des Sciences Exactes et Informatique, université de Jije, 2014.
- [25] BELKHITER, Khalid. «Sélection des services Web à base d’algorithmes génétiques multi-objectifs.» Mémoire, Université Abou Bekr Belkaid– Tlemcen, 2012.
- [26] BENZINA, Youcef, et Hassen BEKKADOUR. «La sélection multi objectif des services web basé de recuit simulé.» Mémoire, Université Abou Bekr Belkaid - Tlemcen, 2012.
- [27] ALILI, Sofiane, et Chrif BASSIM. «La sélection des services Web à base d’algorithme de la recherche Harmonique.» Mémoire, Université Abou Bekr Belkaid– Tlemcen, 2015.
- [29] DAUPHINE, Bouyssou, Marchant THEIRRY, et Patrice PERNY. Théorie du choix social et aide multicritère à la décision. Article, France : Université de Paris , 2015.
- [33] HARRATH, Youssef. Contribution à l’ordonnancement conjoint de la a production et de la maintenance: Application au cas d'un Job Shop. Thèse de doctorat, L’UFR des Sciences et Techniques de l’Université de Franche-Comté, 2003.
- [34] MEHIDID, Fadila. "Algorithme Génétique". Mémoire Fin d'étude, Mostaganem: Univercité Abd Elhamid Ibn Badis, 2013.
- [35] SOUQUET, Amédée, et Francois-Gérard RADET. «ALGORITHMES GENETIQUES.» TE de fin d’année, 2004.
- [36] CHOUCANI, Imed." Utilisation d’un algorithme génétique pour la composition". Mémoire , Canada: univercité de Québec à Montréal , 2010.
- [37] MOULAY KHATIR, Oussama Mohy-eddine, et Redouane HENAOUI. Sélection de Service Web à base. Mémoire, Tlemcen: Université Abou Bakr Belkaid, 2013.
- [38] AIT SI LARBI, Elyassmine. Proposition d'un modele pour la planification de production des entreprises multi site. Doctoral dissertation, Université d'Oran1-Ahmed Ben Bella: , 2009.
- [43] JOURDAN, Cédric. Présentation de JSP et comparaison à PHP. Projet du cours de Systeme de la filière Informatique et Réseaux de 5ème année, Paris: Université de Marne la Vallée.

Rapports techniques

- [7] CHABANA, Refes. Les services web. OpenClassroom, 2016, 321.
- [9] Pillou, Jean-François. WSDL. article, CCM CommentCaMarche, 2008.
- [14] MANI , Anbazhagan. Understanding quality of service for Web services,. IBM, 2002.
- [20] Digital Guide IONOS . Services Web : services entre machines. 2022.
- [39] AltitudeTVMcom. Comprendre NetBeans ainsi que les fonctions, forces et faiblesses de NetBeans. 2020.

- [46] JOHANN, Pardanaud, et Sébastien MARK. «JavaScript.» 2012 .
- [3] Summit, Red Hat. L'architecture orientée services (SOA), qu'est-ce que c'est ? Delarivière Lefouillon France: Red Hat, 27-07-2020.
- [48] VAn, Lancker Luc. HTML: les bases du langage. Belgique: ENI, 2004.

Documents web

- [17] W3C, Groupe. Web Services Architecture. Date de consultation 11-02 -2004.
<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.
- [50] Lechoiscommun. Le paradoxe de Condorcet. Date de consultation 27-04-2016
<https://lechoiscommun.fr/content/article/le-paradoxe-de-condorcet.html>
- [28] MAUDE, Pelletier. procédures de votes - principe condorcet. vidéo Youtube, Date de consultation 10-02-2021.
- [30]Techno-Science.net. Algorithme génétique. Date de consultation 06-06-2004
<https://www.techno-science.net/glossaire-definition/Algorithme-genetique.html>.
- [31] 24PM acadimy A. Igorithme génétique. Édité par Raphaël Richard. Date de consultation 04-10-2019. <https://24pm.com/117-definitions/375-algorithme-genetique> (accès le).janvier 2019. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction/>.
- [40] Clever Cloud. Clever Cloud Technologie. Date de consultation 01-03-2022.
<https://www.clever-cloud.com/fr/technologie/>.

Article d'un ouvrage collectif

- [8] DOUDOUX, jean-michel. «Les services web de type Soap.» Dans Développons en Java, de jean-michel. DOUDOUX. 2019.
- [4] La Rédaction JDN . XML (Extensible Markup Language). Définition et description simple. JDN.2019
- [5] La Rédaction JDN. «HTTP (HyperText Transfert Protocol) définition, traduction.» JDN, 2019.
- [32] Academy by Apside, Performer, *Qu'est-ce que sont les algorithmes génétiques ?*. [Sound Recording]. 2022.
- [41] DOUDOUX, Jean-Michel. «GSON.» Dans Développons en Java. 2022.
- [42] Comment Ça Marche. «JDBC - Introduction.» 2007.
- [44] La rédaction JDN. Java. 2019.
- [45] Acervo Lima. «Avantages et inconvénients de Java.» 2021.
- [47] GeegsForGeeks, Advantages and Disadvantages of javaScript,Noidia,Uttar Pradesh,2021
- [49] J.-M. DOUDOUX, "JSON," in *Développons en Java*, 2022.