République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

MÉMOIRE DE MASTER PROFESSIONNEL

En

Informatique

Option

Administration Et Sécurité Des Réseaux

Thème

Approche de sélection et de composition de services basée sur un algorithme méta-heurstique

Réalisé par :

Mlle. BOUKHAMA Macilia

Mlle. BOULAHOUAT Celia

Evalué le .. octobre 2020 devant le jury composé de :

Examinateur 1	**M. Mir Foudil**	**M.A.A**	**U. A/Mira Béjaia.**
Examinateur 2	**Dr Khanouche M.Essaid**	**M.C.A**	**U. A/Mira Béjaia.**
Rapporteur	**Dr Farah Zoubeyr**	**M.C.A**	**U. A/Mira Béjaia.**

Béjaia, octobre 2020.

⁂ *Remerciements* ⁂

En préambule à ce mémoire nous remercions ALLAH qui nous a donné le courage, la force et la patience d'accomplir ce modeste travail.

Nos remerciements vont en premier lieu à notre encadrant M.FARAH Zoubeyr pour tout le temps qu'il nous a consacré, pour ses précieux conseils, ses encouragement sans lesquels ce travail n'aurait pu voir le jour.

Nous adressons nos sincères remerciements à chacun des membres du jury M.MIR et M.KHANOUCHE pour l'intérêt qu'ils ont portés à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nous remercions nos familles respectives et particulièrement nos parents pour leurs soutiens qu'ils nous ont accordés tout au long de notre chemin.

Nous remercions tout nos enseignants du département informatique de l'université ABDR-RAHMANE MIRA DE BEJAIA.

Nous n'oublions pas de remercier nos amis (e) qui sont toujours présents et fidèles, et tous ceux qui ont contribué, de prés ou de loin à réaliser ce travail.

⁂ *Dédicaces* ⁂

A la mémoire de ma mère NOURIA,

Celle qui m'a donnée la vie , la tendresse et le courage pour réussir.
Aucune dédicace ne saurait exprimer l'amour et la reconnaissance que je te porte.
En témoignage, je t'offre ce modeste travail malgré t'es loin de moi afin de te remercier pour tes sacrifices et ton amour pour moi, et que **ALLAH** t'accueille dans son vaste paradis.

A mon père RACHID,

L'épaule solide, l'œil attentif compréhensif c'est la personne la plus digne de mon estime et de mon respect.
Aucune dédicace ne saurait exprimer mes sentiments, que **ALLAH** te préserve et te procure santé et long vie.

A mes frangins,

En particulier à ma sœur NAWEL pour sa tendresse et ses encouragements, je te souhaite tout le succès...et le bonheur.

A ma petite cousine AMINA BOUKHAMA,

A mes amis,

En particulier BENBEKKA SIHAM, CHERFA DOUNIA, BENKHLAT DRIFA et BENKHLOUF MALIKA qui mon beaucoup aidées à accomplir ce travaille, je vous souhaite tout le succès...et le bonheur.

A ma binôme CELIA et toute la famille BOULAHOUAT,

A toute personne.

Qui m'a aidé à franchir l'horizon...

Boukhama Macilia

⁂ *Dédicaces* ⁂

A mes chères parents,

Sans vos sacrifices, votre tendresse et votre affection je ne pourrais arriver jusqu'au bout j'essayerai toujours d'être à la hauteur de votre confiance.

A mon cher frère,

A mes grands-parents,

A qui je souhaite une longue vie...

A tous mes amies,

A ma binôme MACILIA et toute la famille BOUKHAMA,

A tous ceux que j'aime.

Celia Boulahout

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Liste des abréviations

API	**A**pplication **P**rogramming **I**nterfaces
AS	**S**bstract **S**ervice
B2B	**B**usiness to **B**usiness
B2C	**B**usiness to **C**onsumer
CASP	**C**orrelation **A**ware **S**ervice **P**runing
CASP4AT	**C**orrelation **A**ware **S**ervice **P**runing for **A**djacent **T**asks
CASPN4AT	**C**orrelation **A**ware **S**ervice **P**runing for **N**andjacent **T**asks
CC	**C**oncrete **C**omposition
CPSS	**C**yber **P**hysical **S**ocial **S**ystems
CS	**C**andidate **S**ervice
DPSA	**D**istributed **P**artial **S**election **A**lgorithm
EQSA	**E**nergy-centred and **Q**oS-aware **S**ervices selection **A**lgorithm
ESWOA	**E**agle **S**trategy with **W**hale **O**ptimization **A** lgorithm
FOCC	**F**uzzy **O**ptimal **C**ontinuity **C**onstruction
FWA	**F**ireworks **A**lgorithm
FW-WSC	**F**ireworks algorithm-based QoS-aware **W**eb **S**ervices **C**omposition approach
GA	**G**enetic **A**lgorithm
GWO	**G**ray **W**olf **O**ptimizer
HHO	**H**arris **H**awks **O**ptimization
HTTP	**H**yper **T**ext **T**ransfer **P**rotocol
IBM	**I**nternational **B**usiness **M**achines
LCSDP	**L**ow-Carbon **S**ustainable **D**evelopment
PSO	**P**article **S**warm **O**ptimization
QoS	**Q**uality of **S**ervice

SMT	**S**urface **M**ount **T**echnology
SOA	**S**ervice **O**riented **A**rchitecture
SOC	**S**ervice **O**riented **C**omputing
SOAP	**S**imple **O**bject **A**ccess **P**rotocol
UDDI	**U**niversal **D**escription **D**iscovery and **I**ntegration
URI	**U**niform **R**esource **I**dentifier
W3C	**W**orld **W**ide **W**eb **C**onsortium
WOA	**W**hale **O**ptimization **A**lgorithm
WS	**W**eb **S**ervice
WSDL	**W**eb **S**ervice **A**lgorithm **L**anguage
WSN	**W**ireless **S**ensor **N**etwork
XML	**X**tensible **M**arkup **L**anguage

Introduction Générale

De nos jours, plusieurs services Web sont disponibles sur Internet, offrant des fonctionnalités similaires avec différents niveaux de qualité de service (QoS). Ces services en tant qu'entités individuelles ne peuvent souvent pas satisfaire la demande d'un utilisateur ou de répondre à une requête complexe. Afin de satisfaire une demande d'un utilisateur, plusieurs services Web peuvent être composés, dans processus appelé la composition de services. Ce processus permet de fournir des fonctionnalités qu'aucun service atomique ne peut offrir individuellement. Par ailleurs, la composition de services est un processus qui fait intervenir un autre processus intérimaire appelé la sélection de services. La sélection individuelle des services candidats est effectuée en tenant compte des attributs de Qos de telle sorte que les exigences et les contraintes de l'utilisateur soient satisfaites. Chaque service candidat d'un service abstrait unique possède une valeur de QoS différente permettons de choisir le service adéquat pour le service composite final.

Le problème de composition et de sélection de services a été largement traité dans la littérature, principalement, on trouve des solution déterministes et des solutions non éterministes [ATMANI and CHERIFI, 2018]. Les approches déterministes se basant sur des algorithmes de recherche déterministes tels que dans [Chen et al., 2015, Khanouche et al., 2016, Kumar and Purohit, 2016, Wang et al., 2017, Purohit and Kumar, 2019, Khanouche et al., 2020]. Certaines approches déterministes sont limitées dans les environnements à large échelle avec l'augmentation exponentielle de nombre de services candidats car, elles se basent sur des alorithmes de recherche exacte tels que [Chen et al., 2015] et [Wang et al., 2017] qui nécessitent un temp d'exécution élevé. Les solutions non-déterministes sont des solutions basées sur des algorithmes de recherche méta-heuristiques stochastique tels que dans[Zhao et al., 2017, Karimi and Babamir, 2017, Jatoth et al., 2018, Gavvala et al., 2019, Li et al., 2020]. Les algorithmes méta-heuristiques tentent d'identifier la solution optimale globale en explorant l'espace de recherche d'une façon aléatoire.

Peu d'algorithmes méta-heuristiques peuvent fonctionner correctement pour certaines

classes de problèmes. Cela est dû à l'aspect aléatoire utilisées pour résoudre les problèmes NP-difficiles. Dans certains cas, la diversité de la population est faible, ce qui diminue la possibilité de parvenir à une solution globale, et cela qui entraîne une convergence prématurée de l'algorithme.

Dans ce travail, nous proposons une nouvelle approche de sélection et de composition de services, nommée : Fireworks algorithm-based QoS-aware Web Services Composition Approche (FW-WSC). Approche basée sur l'algorithme des feux d'artifice (FWA), est en mesure de réduire le temps de composition et de satisfaire au mieux les exigences de l'utilisateur en terme de QoS vu que elle assure la diversité de la population car le principe de FWA permet de mettre à jour la population à chaque itération, cela assure une grande capacité d'exploration et permet d'atteindre l'optimum globale.

Ce mémoire, est organisé en quatre chapitres.

Dans le chapitre 1 nous introduisons, les principes de la technologie des services web en définissant le paradigme SOA et les standards de développement des services Web, ensuite nous donnons quelques concepts liés à la composition de services et nous introduisons les processus de sélection et de composition de services.

Dans le chapitre 2, nous étudions tout d'abord et classifions les approches de compositions de services proposées selon leurs méthodes de résolution. Ensuite, nous comparons les approches étudiées selon leurs avantages, leurs limitations et d'autres caractéristiques tels que le passage à l'échelle, l'optimalité et la prise en compte des paramètres de QoS. Enfin, nous terminons ce chapitre par une discussion récapitulative des approches étudiées.

Dans le chapitre 3, nous présentons tout d'abord les modèles de composition avec prise en compte des contraintes de QoS. Par la suite nous présentons l'algorithme de feux d'artifice (FWA)[Tan and Zhu, 2010]. Enfin nous détaillons l'approche de composition de services FW-WSC qui comprend trois phases : une phase d'initialisation, la phase de génération des services composites et la phase de sélection des services composites.

Le chapitre 4 est dédié à l'évaluation des performances de l'approche FW-WSC en termes de temps de composition et de valeur d'utilité.

Nous terminerons ce mémoire par une conclusion et des perspectives futures.

Chapitre 1

Généralités sur SOA et les services web

1.1 Introduction

L'évolution d'applications distribuées dans le monde d'informatique a engendré le développement de l'nformatique Orientée Services (SOC – Service Oriented Computing) qui est un paradigme d'interaction entre services en se basant sur l'architecture orientée services (SOA – Service Oriented Architecture), qui s'articule généralement sur la technologie des services web, cette dernière représente une solution émergentes et prometteuses pour le développement, le déploiement et l'intégration des applications capable de satisfaire les besoins des utilisateurs.

Dans ce chapitre, nous présentons le paradigme l'Informatique Orientée Services (SOC – Service Oriented Computing) et l'architecture orientée service avec quelques-unes de ses caractéristique et ses acteurs. Nous mettons aussi l'accent sur le concept de service ainsi sur ses aspects non fonctionnels notamment les paramètres de qualité de services (QoS). Nous présentons la pile des technologies des services web. Nous terminons le chapitre par la présentation des étapes du processus de la sélection et de la composition de services.

1.2 L'informatique orientée services

L'Informatique Orientée Services (SOC – Service Oriented Computing) est un paradigme de l'informatique distribuée qui utilise les services comme éléments fondamentaux pour le développement d'applications. Pour construire le modèle de service, SOC s'appuie sur l'architecture orientée services (SOA – Service Oriented Architecture), cette derniere est un moyen de réorganiser les applications logicielles et l'infrastructure en un ensemble de services interactifs [Papazoglou, 2003].

1.3 Le concept de service

Un service c'est une entité logicielle qui fournit un ensemble de fonctionnalités accessible via une interface, cette dèrniere contient des informations fonctionnelle du service aussi des informations sur ses aspects non-fonctionnels. A partir de cette spécification, un consommateur de service peut chercher un service qui répend à ses besoins, le sélectionner et l'invoquer selon un contrat de négociation entre le client et le fournisseur qui spécifie la qualité de service fournis et les pénalités encourues en cas de manquement[Aiello et al., 2005] [Chollet, 2009].

1.3.1 Les propriétés fonctionnelles

Les propriétés fonctionnelles d'un service désignent les fonctionnalités que ce service peut fournir. Les propriétés fonctionnelles sont décrites dans la description de service en termes d'opérations, entrées/sorties et post-conditions qui reflètent le fonctionnement du service.

1.3.2 Les propriétés non-fonctionnelles

Les propriétés non-fonctionnelles sont des attributs de qualité de service(QoS) permettant une évaluation des performances d'un service.

1.3.3 Les attributs de QoS

Les attributs QoS c'est des mesures qui permet de quantifier un service ou un service composite. De plus, les QoS représentent l'aspect non-fonctionnel d'un service. Les attributs QoS comprennent : le coût, le temps de réponse, la disponibilité, la réputation, la fiabilité, etc. Les attributs QoS peuvent être divisés en : attributs positifs (par exemple, débit et disponibilité)qu'il faut maximiser et négatifs (par exemple prix et temps de réponse)qu'il faut minimiser. Le tableau (1.1) présente les attributs de qualité de service les plus nombreux dans la littérature [Zeng et al., 2004].

Attribut	**Discription**
Temps de réponse	C'est le délai de délivrance d'un service par un fournisseur en répandant à la requête de consommateur. Le temps de réponse est mesuré en fonction de certains sous-facteurs tels que le temps de réponse moyen et le temps de réponse maximal promis par le fournisseur de services [Garg et al., 2013].
Débit	C'est le nombre de requêtes exécuter par un service dans une duré de temps prédéfinit.
Disponibilité	Désigne la probabilité qu'un service Web soit disponible. Nous définissons un service disponible comme un service capable de répondre à la requête de l'utilisateur dans un délai prédéfinit.
Fiabilité	Cet attribut représente la performance du service à fonctionner correctement et de manière cohérente selon les conditions définies dans le contrat de négociation entre le fournisseur et le consommateur.
Cout	Il se réfère aux frais d'accès et d'utilisation d'un service que le demandeur de service doit payer. En outre, cela dépend du nombre de tâches qu'un utilisateur de service doit exécuter.
Sécurité	comporte différents aspects, tel que la confidentialité, l'intégrité caractérisant la sécurisation des échanges de messages entre le client et le serveur.

TABLE 1.1 – Les attributs de QoS les plus utilisés au niveau applicatif.

1.4 Les principes d'une architecture orientée services

L'architecture orientée service constitue un style d'architecture basée sur le principe de séparation de l'activité métier en une série de services et de les administrer afin de satisfaire les besoins demandé par un consommateur de services, Ces derniers peuvent être assemblés et liés entre eux selon le principe de couplage lâche pour exécuter l'application désirée [Bieberstein et al., 2006].

1.4.1 Les acteurs de SOA

Une architecture orientée service est un paradigme fondée sur la description de services et sur la description de leurs interactions [Dodani, 2004], elle s'articule autour de trois acteurs fondamentaux à savoir le fournisseur de services, l'annuaire de publication et le consommateur de services(client) (voire la figeur 1.1).

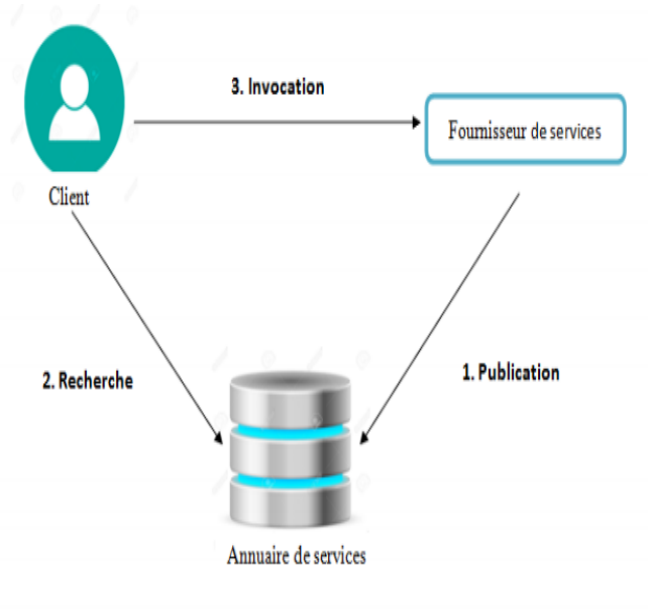


FIGURE 1.1 – Architecture SOA.

- **Le fournisseur de service**

  C'est le propriétaire du service car c'est le responsable du développement du service, de son déploiement, de son exécution et publié toutes ses caractéristiques dans un annuaire de service afin de l'utiliser, ainsi que les opérations disponibles et leur mode d'invocation.
- **L'annuaire de services**

  C'est donc un registre de description qui offre aux fournisseurs le moyen de publier et d'indexer leurs services. Il permet également aux clients de rechercher ces services selon plusieurs critères. Les interactions de base qui existent entre ces trois éléments sont principalement les opérations de : publication, de recherche, d'invocation et de lien ou de liaison (appelé aussi Bind).
- **Le consommateur de service (client)**

  quant à lui, accède à l'annuaire pour rechercher les services dont il a besoin et avec lui les normalisations à obtenir. Il peut s'agir soit d'applications clientes ou bien soit de services qui s'appuient sur la fonctionnalité d'un autre fournisseur de service. Le client peut aussi envoyer ses requêtes au service désiré et obtenir les réponses qu'il pourra analyser.

1.4.2 Les opérations possibles dans une SOA

- **Publication :** pour être accessible, une description de service comportant les propriétés fonctionnelles du service (fonctionnalités) ainsi que ses propriétés non-fonctionnelles doit être publiée de la part du fournisseur de services enregistré auparavant dans un annuaire de services de sorte qu'elle puisse être découverte et appelée par un consommateur de

services.

- **Recherche :** c'est un procédé qui consiste à rechercher dans un registre les services répondant aux critères du service demandé par le client.
- **Invocation :** après recherche de la description de service, le consommateur de services procède par appel du service selon les informations relatives à la description du service.

1.5 Service atomique vs service composite

Un service est dit composite lorsque son exécution nécessite d'invoquer plusieurs services atomiques existant en faisant appel à leurs fonctionnalités afin, de répandre à des fonctionnalités plus complexe. Un service caractérisé par une exécution autonome et qui ne fait pas appel à d'autres services est dit atomique.

1.6 Service abstrait vs service concret

Deux niveaux d'abstraction de services peuvent être distingués : service concret et service abstrait.

1.6.1 Service concret

Un service concret est une fonction qui agit sur des données d'entrée pour retourner des résultats en sortie, après son exécution. Un service concret CS_i est décrit par des propriétés fonctionnelles et non-fonctionnelles (voire la figure 1.2).

Les données d'entrée du service CS_i^{in}, les données de sortie du service CS_i^{out} , les pré-conditions du service $Prec(CS_i)$, et les effets du service $\mathit{eff}(CS_i)$ sont la partie fonctionnelle, et la partie non-fonctionnelle est définie par un ensemble d'information contextuelle du service Contexte (par exemple la localisation du service) et par des attributs de la qualité de service QoS(CS_i)[Yachir et al., 2008].

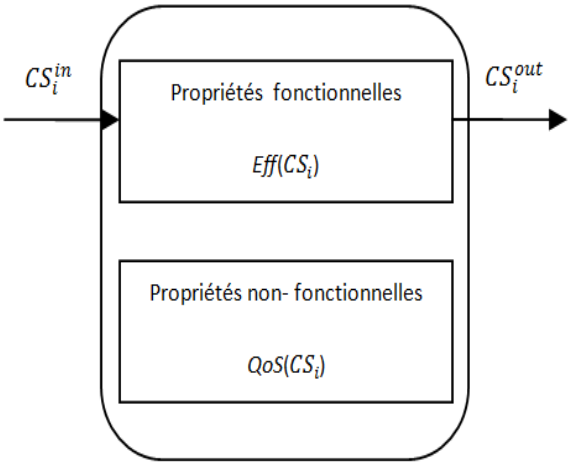


FIGURE 1.2 – Représentation d'un service concret.

1.6.2 Service abstrait

Un service abstrait, représente un ensemble de services concrets qui ont les mêmes fonctionnalités et les mêmes paramètres d'entrées/sorties, un service abstrait AS_i est décrit par des données d'entrées du service $AS_i{}^{\text{in}}$, des données de sortie du service $AS_i{}^{\text{out}}$, et un ensemble de services concrets CS_i (voire la figure 1.3).

Un service abstrait est aussi appelé classe de service, ainsi un service concret peut être vu comme une instance d'un service abstrait [Tari et al., 2009].

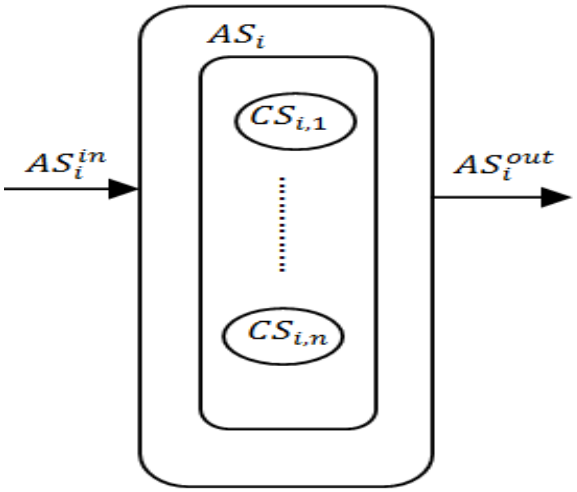


FIGURE 1.3 – Représentation d'un service abstrait.

1.7 Définition des services web

L'activité de standardisation relative aux services Web du consortium W3C définit un service Web comme une application ou un composant logiciel identifié par un URI, ses interfaces et ses liens peuvent être décrits avec le format XML, sa définition peut être découverte par d'autres systèmes logiciels et services Web, il peut interagir directement avec d'autres services Web à travers le langage XML portés par les protocoles Internet.

Les services web ont été proposés initialement par IBM, ils sont définit comme un ensemble de fonctions d'application associées qui peuvent être invoquées via Internet. Les services web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service web est déployé, d'autres applications (y compris des services web) peuvent le découvrir et l'invoquer.

Une définition exacte du terme service Web n'existe toujours pas mais parmi les différentes définitions proposées s'accordent au moins sur l'idée qu'un service Web est un nouveau type de composant logiciel ayant la capacité de publier ses fonctions sur Internet sous forme de services, de rendre ces services facilement invocables et de les mettre à disposition par l'intermédiaire de protocoles Internet standardisés tels que le protocole http qui est largement utilisé par les navigateurs internet ou tout autre protocole comme ceux basés sur le format XML.

Le modèle des services Web repose sur l'architecture orientée service. Celle-ci fait intervenir trois acteurs importants interagissent les uns avec les autres. La Figure1.4 explique les étapes de fonctionnement de l'architecture des services web :

1. Dans un premier temps le fournisseur de service Web publie ses services web.
2. Le client envoie une requête à l'annuaire de Service pour trouver le service Web dont il a besoin.
3. L'annuaire cherche pour le client, trouve le service Web approprié et renvoie une réponse au client en lui indiquant quel serveur détient ce qu'il recherche.
4. Le client envoie une deuxième requête au serveur pour obtenir le contrat de normalisation de ses données.
5. Le serveur envoie sa réponse sous la forme établie par WSDL en langage XML.
6. Le client appel le service web sous la forme établie par SOAP en langage XML.

7. Le serveur fait les calculs nécessaires suite à la requête du client, et renvoie sa réponse sous la même forme normalisée

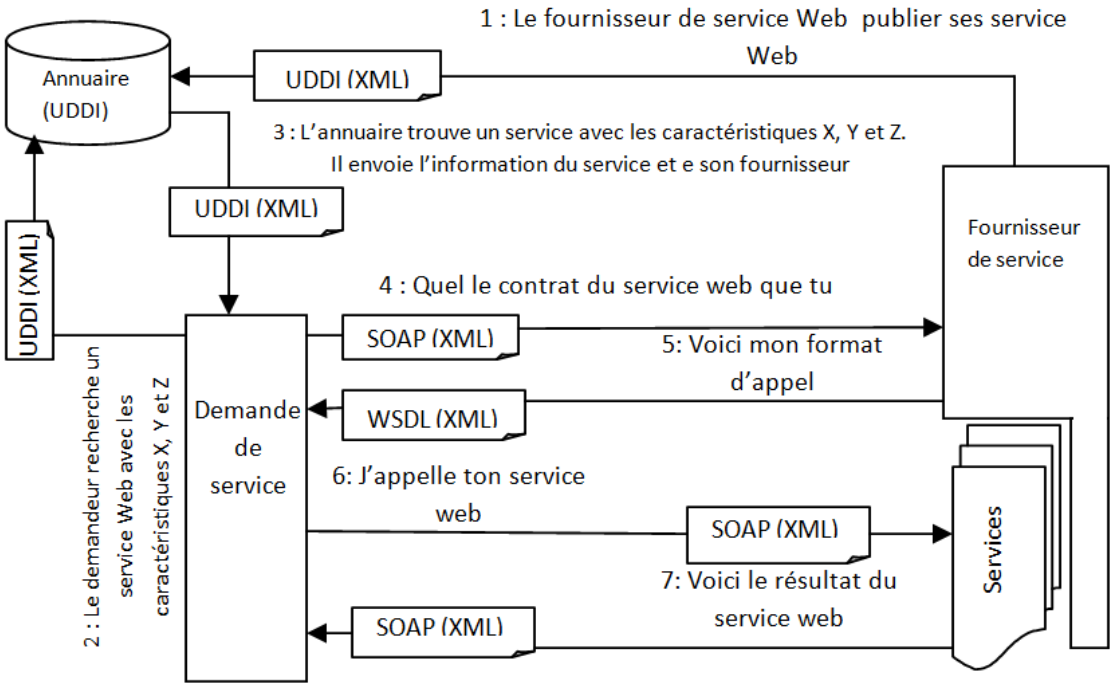


FIGURE 1.4 – Le fonctionnement de l'architecture des services web.
Source : [BEKKOUCHE, 2018].

1.8 Architecture en couches

Différentes extensions de l'architecturer de base ont été proposées par le groupe architecture du W3C qui travaille activement à l'élaboration d'une architecture étendue standard. L'architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de la pile service web. La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base. Ces couches s'appuient sur les standards émergents SOAP, WSDL et UDDI [Kellert and Toumani, 2003] . Plus à 2 autres types de couches supplémentaires :

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

Et une couche Business processus : Permet l'utilisation effective des Web services dans le domaine du e-business. La figure (1.5) décrit un exemple d'une telle pile :

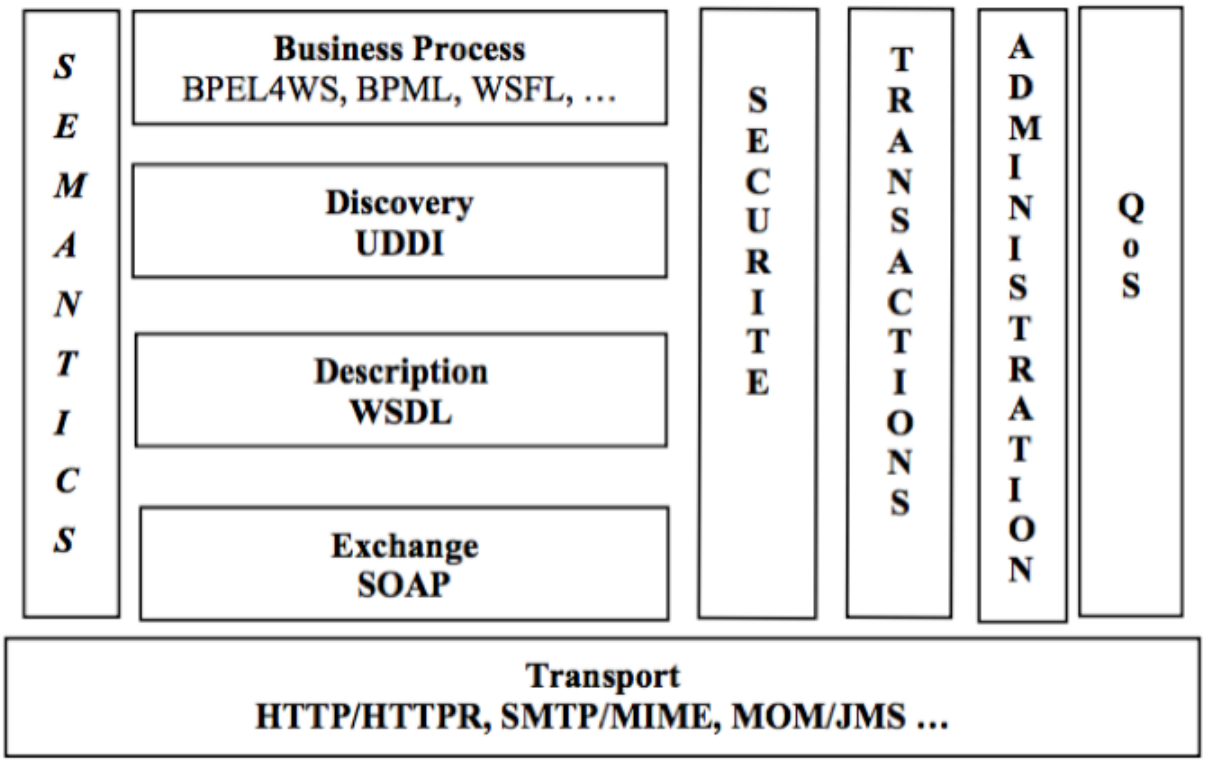


FIGURE 1.5 – L'architecture en couches.
Source : [BENKHALED and REGRAGUI, 2017].

1.9 Les technologies des services Web

Les principaux standards et normes impliqués dans la conception et le développement des services Web sont :

1.9.1 SOAP(Simple Object Access Protocol)

Le protocole Simple Object Access Protocol (SOAP) est un protocole adopté par le consortium W3C dans le but d'échange des données et d'assurer l'interconnexion des Web services en transportant les paquets de données encapsulés. En effet, le protocole SOAP consiste à faire circuler des données encodées généralement en XML, via le protocole HTTP ou SMT avec le port 80 par défaut. Ce choix facilite grandement les communications car le XML est un langage standard et le port utilisé est le port 80 (le même utilisé pour toute navigation Internet sur le Web) ce qui permet de traverser les proxys et les pare-feu (firewalls). Une application (client) envoie une requête SOAP à un Web service, et le Web service retourne la réponse dans ce qu'on appelle une réponse SOAP, ça fonctionne de manière synchrone et asynchrone.

1.9.2 Le langage XML (eXtensible Markup Language)

Le langage eXtensible Markup Language (XML) est un standard promulgué par le consortium W3C qui est l'organisme chargé de standardiser de nombreuses technologies relatives à l'évolution du Web. Il permet de décrire des documents structurés transportables sur les protocoles d'Internet. En effet, il apporte à l'architecture des services web l'extensibilité et la neutralité vis à vis des plateformes et des langages de développement [Yachir, 2014]. De plus, grâce à la structuration, XML permet la distinction entre les données des applications et les données des protocoles.

1.9.3 WSDL(Web Service Description Language)

Le langage de description WSDL se comporte donc comme un langage permettant de décrire l'interface visible (ou publiée) du service web. Il décrit, à l'aide du langage de balises XML, les différents éléments du service. Un fichier WSDL contient principalement sept éléments (voir Figure1.6) sont :

- **DataTypes :** fournit la définition des types de données utilisés pour décrire les messages échangés par le service web.
- **Messages :** représente une définition abstraite (noms et types) des données en cours de transmission.
- **PortType :** est un groupement logique ou une collection d'opérations supportées par un ou plusieurs protocoles de transport, il est analogique à une définition d'un objet contenant un ensemble de méthodes.
- **Binding :** Décrit la façon dont un type de port est mis en œuvre pour un protocole particulier (HTTP par exemple), et un mode d'invocation (SOAP par exemple). Cette description est faite par un ensemble donné d'opérations abstraites. Pour un type de port, on peut avoir plusieurs liaisons, pour différencier le mode d'invocation ou de transport de différentes opérations.
- **Service :** indique les adresses de port de chaque liaison.
- **Port :** représente un point d'accès de services défini par une adresse réseau et une liaison.
- **Opération :** c'est la description d'une action exposée dans le port.

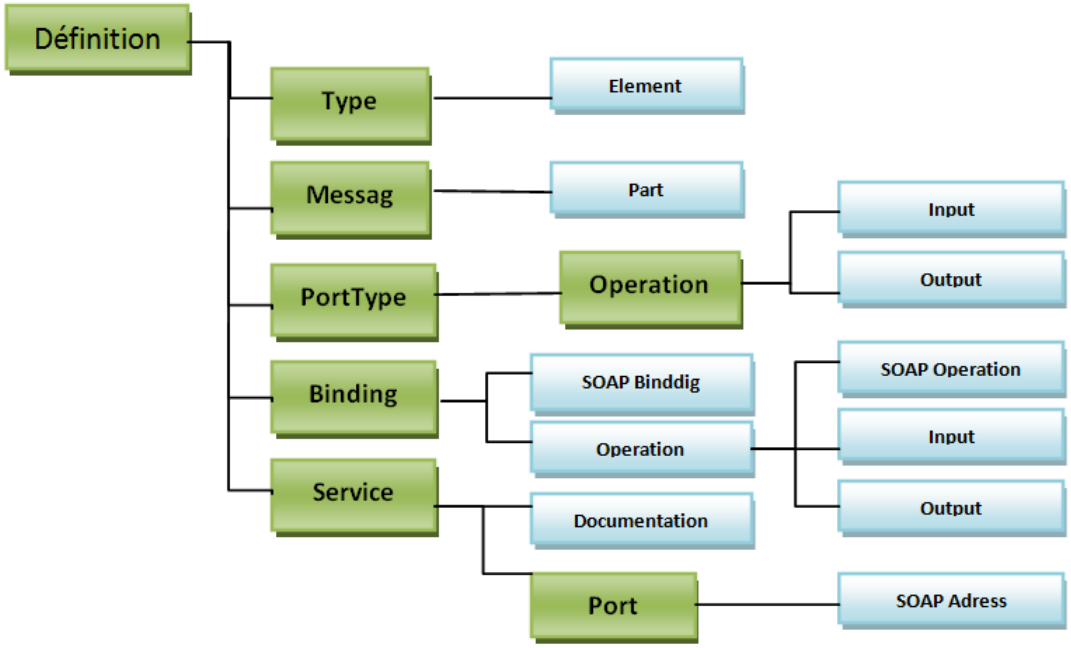


FIGURE 1.6 – Les éléments d'un fichier WSDL.

1.9.4 UDDI(Universal Description Discovery and Integration)

UDDI est introduit en 2000 par Ariba, Microsft et IBM, il n'est pas gérée par le W3C mais par le groupe OASIS. Il a été crée pour la publication et la découverte des informations sur les services Web. La spécification UDDI vise à créer une plateforme indépendante, un espace ou cadre de travail (framework) ouvert pour la description, la découverte et l'intégration des services des entreprises.

1.10 Les application des services Web

Les technologies des services Web peuvent être appliquées à toutes sortes d'applications auxquelles elles offrent des avantages considérables en comparaison aux anciennes API. Les applications des services Web sont multiples, autant dans les domaines du Business to Consumer (B2C), Business to Business (B2B) que dans des domaines de gestion.

- **B2C (Business to Consumer) :** qualifie une application, un site Internet destiné au grand public.
- **B2B (Business to Business) :** qualifie une application, un site Internet destiné au commerce de professionnel à professionnel.

1.11 Sélection de services

La repense aux requêtes complexe nécessite l'intervention de plusieurs services qui existaient déjà (la composition de services), alors la problématique posé : *«comment choisir efficacement des services parmi un ensemble de services fonctionnellement équivalent, sur la base des propriétés fonctionnelles et non-fonctionnelles ? »* [Canfora et al., 2005]. En génirale, c'est compliquer de choisir des services qui répondent aux besoins des utilisateurs [Yachir, 2014]. Un processus de sélection de services est mis en œuvre, qui consiste a choisir des candidats dans chaque ensemble de services. Le processus de sélection est exécuté en eux étapes : 1)la découverte des services publiés dans un répertoire de services, cette étape consiste de faire une correspondance fonctionnelle entre les services abstraits composites et les services disponible afin d'obtenir des services concret pour chaque service abstrait participent à la composition. 2)la sélection des services basée sur propriétés non fonctionnelles, elle consiste à choisir un service concret candidat de chaque service abstrait avec la meilleur Qos qui satisfait les exigence des utilisateurs, comme le montre la figure (1.7).

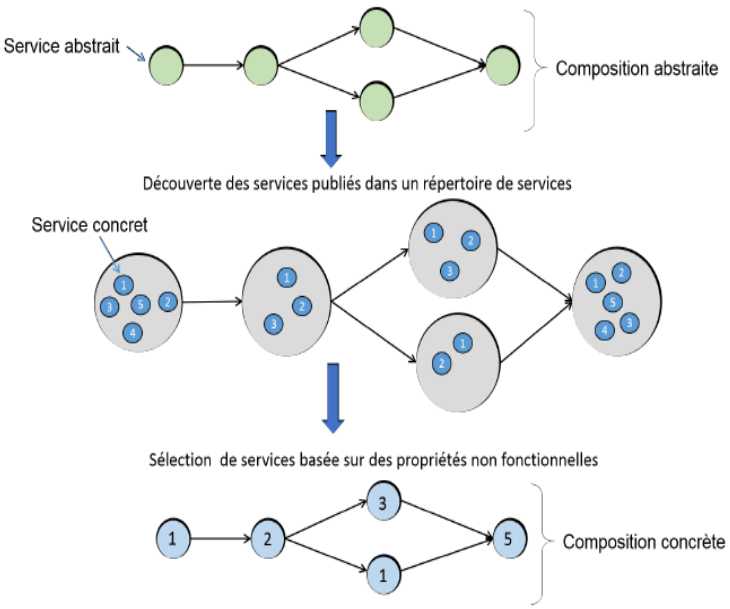


FIGURE 1.7 – Le processus de sélection de services.

1.12 Les types de sélection de services

Sélection locale : La sélection locale son but est de trouver le meilleur service pour chaque tâche individuelle en considérant des contraintes de QoS relatives à chaque sens prendre en

compte des contraintes de QoS globales exprimées pour l'ensembles des tâches. Il s'agit de sélectionner pour chaque tâche un service apte à l'exécuter en tenant en compte les contraintes de l'utilisateur.

Sélection globale : Contrairement à la sélection locale , cette stratégie consiste à choisir une combinaison de services parmi toutes les combinaisons possibles, qui satisfait la contrainte globale exigé par un utilisateur. afin d'appliquer cette méthode une valeur basé sur QoS se fait en utilisant la fonction d'agrégation et la composition qui offre la meilleur valeur sera choisi pour satisfaire la contrainte globale.Ce type de sélection réduit le temps d'exécution.

1.13 La composition des services

La composition des services c'est la combinaison de plusieurs services atomique existant pour fournir de nouvelles fonctionnalités qu'un autre service ne peut pas fournir individuellement. Le nouveau service résultant d'une composition de services est appelé service composite. Son exécution nécessite alors l'invocation de plusieurs autres services afin de faire appel à leurs fonctionnalités. Les services invoqués lors d'une composition de services sont appelés services composants.

1.14 Processus de composition de services web

Le cycle de vie de ce processus inclut cinq phases : comme le montre la figure(1.8).[Moghaddam and Davis, 2014].

1. **Spécification de l'objectif :**Au cours de cette phase, l'objectif et les préférences du client sont définis. Ensuite, l'objectif est décomposé semi-automatiquement en un processus d'affaire abstrait(BP-abstract business process) qui contient un ensemble de services abstraits avec la description de ces fonctionnalités ainsi, le contrôle et le flux de données entre eux.
2. **La phase de découverte :** Dans le registre de services une recherche est effectuée afin de localiser des services web concrets qui correspondent aux exigences fonctionnelles et non fonctionnelles des services abstrait. par conséquent des services web candidats ayant des fonctionnalités similaires mais différents par leur niveau de qualité sont trouvés pour caque tache.
3. **La phase de sélection :** Au cours de cette étape, une variété d'algorithme de recherche sont proposées qui aide le demandeur de service à sélectionner les services web qui

correspond avec les services abstrait.

4. **La phase d'Exécution :**Au cours de la phase d'exécution du service, une instance de processus est créée en exécutant le service composite.
5. **La phase de maintenance et surveillance du service :** L'instance de processus serait surveillée en permanence pour d'autres réponses en cas d'échec ou de changement de son statut au stade final du WSC.

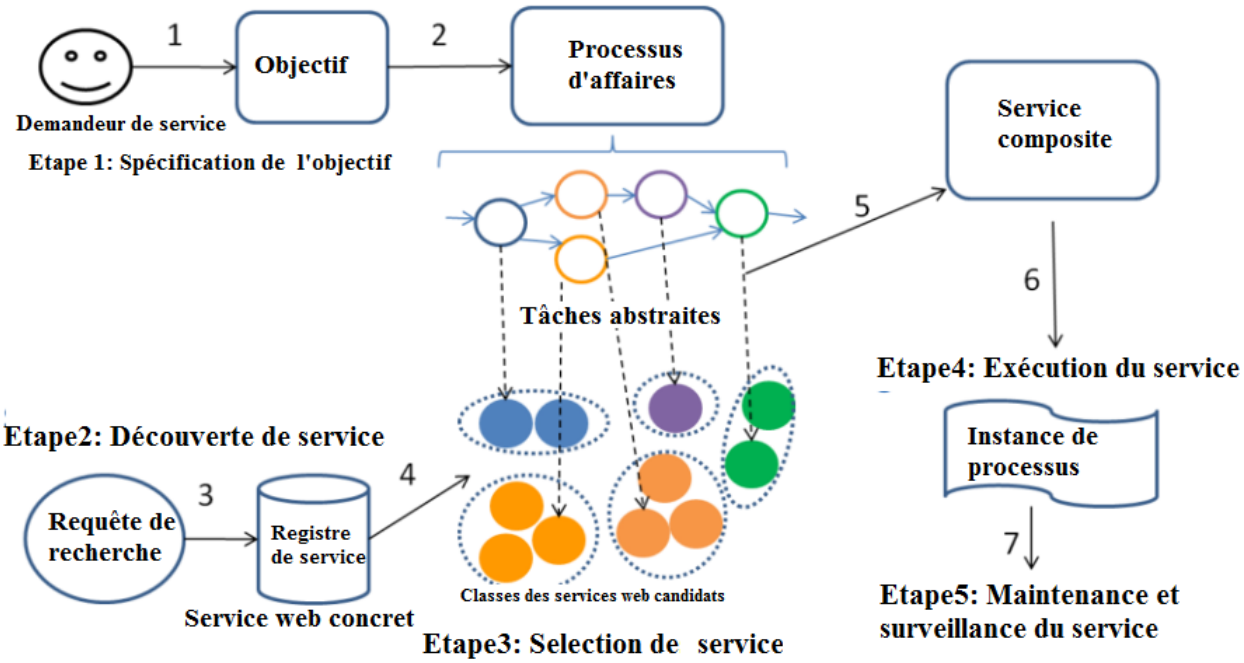


FIGURE 1.8 – Le processus de composition de services web.
Source : [Moghaddam and Davis, 2014].

1.15 Conclusion

La technologie des services web permet à des applications d'interagir entre eux à distance via Internet, indépendamment des plates-formes et des langages sur lesquels elles sont implémentées. En s'appuyant sur des protocoles standards.

Dans se chapitre nous avons présenté de manière générale quelques concepts liés à la technologie des services web. Nous avons d'abord présenté quelques définitions de ce concept. Nous avons également décrit l'architecture fondamentale des services web ainsi que les différents standards proposés par W3C qui permettent de décrire les services web. Cette étape de description est le premier processus indispensable dans le cycle de vie d'une application basée sur une Architecture Orientée Service(SOA). Les services web sont utilisés dans différents domaines

ce qui nécessite la composition de ces services afin de répandre aux besoins complexe de l'utilisateur. Les définitions générales de la composition de services ainsi que la sélection de services sont présentées dans ce chapitre.

Dans le chapitre suivant, nous aborderons le concept de la composition et la sélection de services web ainsi qu'un état de l'art sur les approches de sélection et de composition de services sensible à la QoS.

Chapitre 2

Etat de l'art sur la composition de services web

2.1 Introduction

Dans ce chapitre, nous synthétisons quelques approches de sélection et de composition de services sensible à la qualité de service proposés dans la littérature. Nous commençons par donner quelques concepts liés au problème de sélection et de composition des services web sensible à la QoS. Après cela nous classons les approches de composition de services en deux catégories principales selon leurs techniques de résolution, déterministe ou stochastique comme dans [ATMANI and CHERIFI, 2018]. Ensuite, les approches présentées seront analysées et comparées par rapport aux critères les plus représentatifs de la composition des services.
Nous donnons également les avantages et les limites de chaque approche.

2.2 Modélisation de la composition des services

S= (T_1,...,T_i,...,T_m) représente un service composite où T est une tâche et m c'est le nombre de tâches nécessaire pour former le service composite S. T_i désigne le $i^{\text{ème}}$ service abstrait de la composition S, alors il peut être définit aussi comme suit S=(AS_1,...,AS_i,...,AS_m). AS_i=(WS_1^i,...,WS_j^i,...,WS_n^i)où WS_j^i et $j^{\text{ème}}$ services concret candidat du $i^{\text{ème}}$ service abstrait et n le nombre de services concrets. Les candidats appartenant au même service abstrait on la même fonctionnalité mais différentes valeur QoS. Chaque service candidat WS_j^i est représenté par un vecteur de QoS :QoS(CS_j^i)=($QoS_{1,j}^i$,.., $QoS_{q,j}^i$,..,$QoS_{k,j}^i$) où ($1 \leq q \leq k$) est le nombre d'attributs de QoS et $QoS_{q,j}^i$ représente la valeur du $q^{\text{ème}}$ attribut de QoS du $j^{\text{ème}}$service candidat dans le $i^{\text{ème}}$ service abstrait.

2.2.1 La composition des services Web

La composition de services S=(AS_1,...,AS_i,...,AS_m) est formé de m services abstraits combinés en utilisant une fonction d'agrégation de déférent structures (boucle, séquentielle,parallèle,ect). La composition de services fait appel aux processus de sélection affin de choisir un service concret pour un service abstrait. La combinaison de tout les services sélectionnés dans chaque service abstrait forme un service composite SC_l=(WS^1_{l,j_1},...,WS^i_{l,j_i},...,WS^m_{l,j_m}) où ($1 \leq j_i \leq m$). Ce service composite doit satisfaire les contraintes globales de QoS de l'utilisateur. La qualité de service globales est représenté par un vecteur QoS(SC_l)=($Q_{1,l}$,..,$Q_{q,l}$,..,$Q_{k,l}$) où $Q_{q,l}$ est la valeur agrégée du $q^{\text{ème}}$ attribut de QoS qui est calculé selon la structure de la composition présentée dans le (tableau 2.1) [Sun and Zhao, 2012],$QoS_{q,j}$ représente la $q^{\text{ème}}$ valeur d'attribut QoS du service candidat WS_j,p_j est la probabilité d'exécution d'une branche conditionnelle, m est le nombre de services abstraits dans la composition et y est le nombre de boucles lors de l'exécution d'un service donné .

Type d'agrégation	**Les fonction d'd'agrégation selon la structure de composition**	
	structure de composition	fonction d'd'agrégation
Addition	Séquentiel	$\sum_{j=1}^{m} qos_{q,j}$
	Parallèle	$\sum_{j=1}^{m} qos_{q,j}$
	Conditionnel	$\sum_{j=1}^{m} qos_{q,j} * p_j$
	Boucle	$y * \sum_{j=1}^{m} qos_{q,j}$
Multiplication	Séquentiel	$\prod_{j=1}^{m} qos_{q,j}$
	Parallèle	$\prod_{j=1}^{m} qos_{q,j}$
	Conditionnel	$\sum_{j=1}^{m} qos_{q,j} * p_j$
	Boucle	$qos_{q,j}^{y}$
Max/Min	Séquentiel	$\sum_{j=1}^{m} qos_{q,j}$
	Parallèle	$max_q qos_{q,j}$
	Conditionnel	$\sum_{j=1}^{m} qos_{q,j} * p_j$
	Boucle	$y * \sum_{j=1}^{m} qos_{q,j}$

TABLE 2.1 – Les fonctions d'agrégation des valeurs de Qos selon les structures d'exécution.

Dans notre approche proposé, la fonction d'agrégation des attributs QoS basé sur le modèle de combinaison de séquences, car les autres modèles peuvent tous êtres transformé en modèle séquentiel de manière correspondant [Zeng et al., 2004].

2.2.2 La fonction d'utilité

La fonction d'utilité F est utilisée pour évaluer la qualité d'une composition de services S :

$$F(S) = \sum_{k=1}^{r} UniQ_k * W_k \tag{2.1}$$

où $UniQ_k$ est la valeur normalisée entre $[0, 1]$ du K$^{\text{ème}}$ attribut de QoS du service composite S et W_k est le poids du $K^{\text{ème}}$ attribut de QoS tel que $\sum_{k=1}^{r} W_k = 1$ où W_k est fixé par l'utilisateur selon ses préférences dans l'intervalle $[0, 1]$.
Le calcule de la normalisation se fait selon [Ardagna and Pernici, 2007] comme suit :

$$UniQ_k = \begin{cases} \dfrac{Q_k^{max} - Q_k}{Q_k^{max} - Q_k^{min}} & \text{si l'attribut est négatif} \\ \dfrac{Q_k - Q_k^{min}}{Q_k^{max} - Q_k^{min}} & \text{si l'attribut est positif} \\ 1 & \text{si } Q_k^{max} = Q_k^{min} \end{cases} \tag{2.2}$$

Où Q_k^{max} et Q_k^{min} représentent la valeur maximale et la valeur minimale respectivement du k$^{\text{ème}}$ attribut de QoS de tout les services candidats. parce que les valeurs de qualités des services ont des échelles différentes alors, la normalisation est utilisée comme une forme de standardisation pour transformer le problème de composition de services d'un problème multi-objectif à un problème mon-objectif.

2.3 Taxonomie des approches de composition des services

Les approches de composition de services web sensible à la qualité de service proposées dans la littérature sont classées selon la méthode d'exploration de l'espace de recherche des compositions, en deux catégories principales [ATMANI and CHERIFI, 2018] (voir le tableau 2.2) :

1. Approches déterministes.

2. Approches stochastiques.

Modèle de résolution			Les approches
Déterministes	Autres	Basées sur la corrélation	[Deng et al., 2014]
		Basées sur le clustering	[Kumar and Purohit, 2016]
			[Purohit and Kumar, 2019]
	Basées sur la dominance au sens de pareto		[Chen et al., 2015]
			[Khanouche et al., 2016]
			[Wang et al., 2017]
			[Khanouche et al., 2020]
Stochastiques	Métaheuristique		[Karimi and Babamir, 2017]
			[Zhao et al., 2017]
			[Jatoth et al., 2018]
			[Gavvala et al., 2019]
			[Li et al., 2020]

TABLE 2.2 – Classification des approches de composition de services Web sensible à la qualité de service.

2.3.1 Approches déterministes

Avec un ensemble de données en entrée, un algorithme déterministe exécutera toujours la même suite d'opérations et produira toujours la même sortie, c'est à dire que les approches déterministes parcourent toujours l'espace de recherche des services candidats de la même façon et produisent le même service composite comme résultat finale.

2.3.1.1 Approches basées sur la dominance au sens de pareto.

[Chen et al., 2015, Khanouche et al., 2016, Wang et al., 2017, Khanouche et al., 2020] ont résolu le problème de composition de services sensibles à la QoS en utilisant le concept de dominance au sens de pareto.

1. **Algorithme basé sur la sélection partielle.**

   Dans [Chen et al., 2015], un algorithme DPSA (Distributed Partial Selection Algorithm) qui utilise l'approche de sélection partielle ainsi que le parallélisme est proposé afin de réduire l'espace de recherche ainsi le temps d'exécution. Le DPSA suit le processus de validation des contraintes locales de QoS cela permet l'élimination de tous les services candidats qui violent les contraintes de Qos, puis tout les services non promoteurs qui sont dominés par d'autres de la même classe seront éliminer, cette dominance

est obtenue grâce a un processus de comparaison en formant un ensemble de pareto optimale de services à partir duquel l'ensemble de solutions est déterminé.

Discussion et analyse

Cet algorithme permet de réduire l'espace de recherche en supprimant les services candidats peut promoteurs en termes de Qos. Cependant, il utilise la comparaison par pair ce qui prend du temps en cas d'augmentation des services candidats, ne prend pas en considération la fluctuation de la Qos et ne permet pas d'établir un ordre stricte entre les services candidats.

2. **Algorithme de sélection de service centré sur l'énergie sensible à la Qos.**

Dans [Khanouche et al., 2016], l'algorithme EQSA (an energy-centred and QoS-aware services selection algorithm) est proposé pour réduire la consommation d'énergie d'un service composite à large échelle sans affecter la satisfaction de l'utilisateur en termes de Qos. L'Algorithme EQSA comprend deux phases importantes :

(a) **La phase de présélection :** Cette phase est modélisée comme un problème d'optimisation lexicographique qui consiste à résoudre une séquence de sous problèmes d'optimisation mono objectifs f_n traitant n attributs de Qos sachant que les attributs sont classées dans l'ordre d'importance selon les préférences des utilisateurs, pour chaque sous problèmes f_n on cherche une solution optimale x_n unique, si la solution est trouvée le processus de recherche s'arrête si non on applique une relaxation à la contrainte d'égalité associé à l'objectif c.à.d. dans le cas ou on arrive au dernier objectif sans trouver de solution on lui associe une valeur de seuil de qualité, alors les services candidats qui ne satisfont pas cette valeur sont filtrés et ne participent pas au processus de composition.

(b) **La phase de sélection de services basés sur la dominance relative :** Cette phase consiste à calculer une valeur d'utilité qui dépond du profil d'énergie, les valeurs des attributs de Qos, et les préférences des utilisateurs. La valeur d'utilité est comparée par la suite avec les services présélectionnés dans la phase précédente. Les meilleurs services seront sélectionnés pour la composition de services.

Discussion et analyse

L'algorithme proposé réduit le temps de sélection comme il diminue la consommation d'énergie sans affecter la satisfaction du consommateur. Cependant les services sélectionnés satisfont uniquement les contraintes de Qos locale sans prendre en considération

les contraintes de Qos global.

3. **Composition des services dans le système cyber-physique sociaux.**

Dans [Wang et al., 2017], une approche pour les réseaux physiques, le cyberspace et les réseaux sociaux CPSS (cyber-physical-social systems) également connue sous le nom deFRSkyline est proposée afin de réduire le temps et assurer la fiabilité. Cette approche est une contribution de trois études :

(a) **La dominance de pareto :** consiste à trouver l'ensemble de pareto optimale en terme de Qos on compare les services, et on garde pour chaque service abstrait de la composition, uniquement les services candidats qui ne sont dominés par aucun autre composant, cela permet de réduire l'espace de recherche.

(b) **Le calcule de la fluctuation de Qos :** dans cette étape le coefficient de variation est adopté pour filtrer les composants à fort fluctuation.

(c) **La composition :** enfin l'algorithme MIP (Mixed Integer Programming)est utilisé pour sélectionner les composants les plus fiable avec un cout de temps inférieur pour les utilisateurs des CPSS grâce à l'utilisation d'un moteur de composition [Wagner et al., 2011].

Discussion et analyse

L'algorithme proposé réduit le temps de recherche, assure la fiabilité et prend en considération la fluctuation de la qualité des services. Cependant cette approche peut connaitre un échec lorsque le nombre de composants est faible, le calcule de fluctuation de qualité de service ne peut pas être effectuer pour le prix et ne supporte pas de nouveaux candidats ou bien ceux rarement utilisés.

4. **Composition de services flexible sensible à la qualité de services pour les environnements informatiques de services.**

En effet, l'évolution rapide des environnements orientés services a engendré un accroissement important de services fonctionnellement équivalents. Lorsqu'une composition de services s'impose, il subsiste un dilemme qui a longtemps été étudié dans la littérature : comment choisir des services, fonctionnellement équivalents, qui seraient plus appropriés à la composition et répondent au mieux aux exigences de l'utilisateur en termes de qualité de service. Dans ce context[Khanouche et al., 2020] ont proposés un algorithme de

composition de services sensible à la QoS flexible (FQSC – Flexible QoS-aware Services Composition algorithm) pour augmenter la faisabilité de la composition et réduire le temps de composition, tout en maximisant l'optimalité de la composition en termes de QoS. L'algorithme FQSC pallie les limitations des approches de composition de services basées sur la dominance au sens de pareto et celles des approches basées sur la décomposition des contraintes globales de QoS. En effet, le nombre de compositions augmente de manière exponentielle avec le nombre de services et d'attributs de QoS dans le cas des approches de composition basées sur la dominance au sens de pareto, alors que les approches basées sur la décomposition des contraintes peuvent entraîner une réduction de la faisabilité de la composition dans le cas de contraintes globales de QoS très restrictives. Les résultats de la simulation montrent que l'algorithme FQCA est efficace par rapport à d'autres approches de composition de services en termes de temps de composition, d'optimalité et d'espace de recherche des compositions. .

Discussion et analyse

L'idée principale de l'approche proposée est de rendre les contraintes de qualité de service de l'utilisateur plus flexibles afin d'augmenter la faisabilité de la composition alors que l'espace de recherche de la composition est réduit afin de diminuer le temps de composition, tandis que l'optimalité est maintenue à une valeur élevée. Cependant cette approche ne prend pas en considération les environnements dynamiques.

2.3.1.2 Autres approches

i. Les approches basées sur le clustering·

[Kumar and Purohit, 2016, Purohit and Kumar, 2019] ont résolu le problème de composition de services sensibles à la QoS en utilisant le concept de clustring pour déterminer les services partageant des propriétés similaires et les regrouper.

1. **Explorer le clustering *K-Means* et le Skyline pour la sélection de services Web.**

Dans [Kumar and Purohit, 2016] proposent une solution qui utilise un modèle de sélection de services Web à deux couches :

(a) **La couche de pré-filtrage :** Consiste à utiliser la technique de clustering *K-Means* basée sur les centroïdes afin de générer des groupes de services Web. Les services appartenant au même groupe ont des informations de qualité de services similaires.

Pour les services Web, m nombre de paramètres de QoS sont utilisés pour obtenir le centroïde. Pour générer des clusters de services Web, la technique de clustering *K-Means* prend k en entrée et génère k clusters. Les clusters de services Web sont formés de telle sorte que la similitude de QoS intra-cluster des services Web est élevée et la similitude de QoS intercluster est faible. Le processus commence par considérer les WS individuels comme un cluster et leurs informations QoS comme un centre de cluster. Lors de chaque itération de l'algorithme, la distance euclidean entre deux clusters est obtenue. Deux grappes à la plus petite distance sont fusionnées. Le centre du cluster est réévalué en tant que moyenne de QoS des WS dans ce cluster. Le processus se poursuit jusqu'à ce que la fonction critère atteigne la valeur optimale.

(b) **La sélection du service Web basée sur Skyline :** Système proposé où la couche de sélection de services Web se compose d'un algorithme Skyline. Le paradigme BNL est utilisé pour mettre en œuvre le concept Skyline. La tâche de l'algorithme Skyline est de sélectionner un service Web non dominé. Un service Web S_i est censé dominer le service Web S_j, si toute les attributs QoS de S_i sont meilleurs ou égale à celles de S_j et qu'un seul paramètre de qualité de service est meilleur [Yu and Bouguettaya, 2013] [Alrifai et al., 2010]. L'ensemble non dominé est l'ensemble composé de services Web qui ne sont dominés par aucun autre service Web. Dans chaque itération de l'algorithme, les services Web S_i et S_j sont comparés sur les paramètres QoS. Le service Web qui domine est inclus dans l'ensemble Skyline.

Analyse et discussion

L'approche proposée utilise une architecture à deux couches pour la sélection des services Web. L'utilisation de la technique de clustering *K-Means* pour regrouper les services Web avec une qualité de services (QoS) similaire permet de réduire l'espace de recherche ,ensuite la tâche de sélection de services Web est effectuée sur les services Web résultants en utilisant le concept skyline basé sur l'algorithme BNL qui détermine l'ensemble dominé de services Web. cependant l'algorithme *k-means* n'est pas fiable lors de l'augmentation des services web candidats car le calcule de cette algorithme est limité.

2. **Sélection de services Web à l'aide de calcule skyline.**

Dans [Purohit and Kumar, 2019] une approche basée sur le clustering pour améliorer l'efficacité du processus de sélection en prenant compte les exigences des utilisateurs

finaux de Qos est proposée. Cette approche consiste à appliquer une technique de regroupement (clustering) hiérarchique afin de déterminer les WS candidats partageons des propriétés similaires, cette étape permet la réduction de l'espace de recherche par la suite la méthode de sélection basée sur skyline est utilisée pour sélectionner les services web candidats les plus appropriés (non dominé) pour le processus de composition.

Analyse et discussion

Cette approche utilise la méthode de regroupement des services candidats avec des propriétés fonctionnelles similaire afin de réduire l'espace de recherche, comme la technique de regroupement est hiérarchique alors elle est plus fiable que le regroupement *K-means*. Cependant le calcul de skyline prend beaucoup de temps.

ii. Les approches basées sur la corrélation

Les corrélations de qualité de services c'est un ensemble d'attributs de QoS d'un service qui ne dépendent pas seulement du service lui-même, mais sont également corrélés à d'autres services. [Deng et al., 2014] ont résolu le problème de composition de services sensibles à la QoS en se basant sur la corrélation.

1. **Sélection de services pour la composition avec des corrélations QoS.**

Dans [Deng et al., 2014] une approche avec laquelle ils ont étudiés la façon de sélection de services candidats appropriés tout en tenant compte des corrélations de Qos optimales est proposée. A cette fin ils ont proposés une nouvelle méthode de sélection de services, appelée méthode d'élagage des services sensible à la corrélation (CASP) , afin de réduire la complexité temporelle du processus de sélection des services. Cette approche consiste le prétraitement des ensembles de services candidats en appliquant l'algorithme P4SC qui est divisé en deux étapes :

(a) La première étape consiste à trouver les services avec la valeur de Qos par défaut optimale et à supprimer tous les autres services sans corrélation.

(b) Deuxième étape dans certains tache s'il y'a plus d'une corrélation entres les services candidats correspondants seul celui avec une valeur des Qos composite optimale est réservé et les autres seront supprimés, ce qui permet considérablement de réduire l'espace de recherche et démineur la difficulté de générer les services composites optimaux.

Ensuite l'approche CASP est mis en œuvre afin d'obtenir le service composite le plus optimale. Les corrélations de QoS sont divisées en deux types : 1)les services abstraits adjacents alors on applique l'algorithme CASP4AT (Algorithm for Service Selection with Correlations in Adjacent Tasks) et 2) les services abstraits non adjacents on applique l'algorithme CASP4NAT (Algorithm for Service Sélection with Correlations in Non-adjacent Tasks). Ces deux algorithmes utilisent le concept de préfixe du service composite optimal et élaguent les services composites lorsqu'ils ne sont pas déterminés comme le préfixe du service composite optimal. Les services composites dans les algorithmes (CASP4AT) et (CASP4NAT) sont générés et évalués étape par étape jusqu'à ce que le dernier service abstrait est atteint, à chaque fois qu'un service composite est obtenu , il est considéré comme la composition optimale actuelle.

L'algorithme CASP4AT commence par le premier service abstrait en composant le service composite optimal actuel avec tous les services qui sont en corrélation avec les services suivants et les ajoute à *CorCWSSet*. Ensuite, il choisit le service avec la valeur de QoS par défaut optimale, il le compose avec le service composite optimal actuel et le définit comme le nouveau service composite optimal actuel. Chaque service composite de *CorCWSSet* est composé avec des services qui lui sont corrélés. Le nouveau service composite sera ensuite comparé avec le service composite optimal actuel, le meilleur sera considéré le nouveau service composite optimal actuel et l'autre sera supprimé. Si le service corrélé est également corrélé avec d'autres services dans le service abstrait suivant, le service composite sera ajouté à *CorCWSSet*. L'algorithme avancera comme ceci étape par étape jusqu'à ce qu'il renvoie le service composite optimal actuel comme résultat.

Dans l'algorithme CASP4NAT, pour le premier service abstrait, tous les services candidats trouvés dans le processus de sélection sont composés avec le service candidat, du service abstrait suivant, avec une valeur de qualité de service optimale. Chaque composition de services est composée avec des services candidats en corrélation avec lui. S'il existe deux compositions de services avec le même ensemble de services candidats qui se trouvent dans ces compositions, la composition avec la valeur de qualité de service la plus élevée sera supprimée car elle ne peut pas être le préfixe de la composition de services optimale. Ce processus sera répété jusqu'à ce que le dernier service abstrait soit atteint, et seule la composition de services avec une valeur de QoS optimale sera renvoyée.

Analyse et discussion

Cette approche permet de réduire l'espace de recherche ainsi la complexité temporelle du processus de sélection, puisqu'elle supprime tous les services candidats et les compositions avec des valeurs de QoS faibles et tous les services sans corrélation et peut améliorer la QoS des services composites générés en prenant en compte les corrélations de QoS dans le processus de sélection de services. Cependant, cette approche ne considère que les plans de services avec structure séquentielle. Par conséquent, si la structure du plan de services est plus complexes alors les résultats ne seront pas performant.

2.3.2 Approches stochastiques

Une méthode stochastique parcourt l'espace de recherche de façon « aléatoire ». Deux exécutions successives peuvent donner deux résultats différents, c'est à dire qu'une approche stochastique consiste à explorer l'espace de recherche des services candidats aléatoirement afin d'obtenir une composition.

2.3.2.1 Approches basées sur des méta-heuristiques bio-inspirées

Plusieurs approches de composition de services proposées dans la littérature se basent sur des méta-heuristiques bio-inspirées tel que [Karimi and Babamir, 2017, Zhao et al., 2017, Jatoth et al., 2018, Gavvala et al., 2019, Li et al., 2020]

1. **Composition de service Web sensible à QoS à l'aide de Gray Wolf Optimizer.**

   Dans [Karimi and Babamir, 2017] une approche pour optimiser la composition des services Web sensibles à la QoS à l'aide de GWO (Gray Wolf Optimizer) est proposée. Cette approche consiste à suivre quatre étapes principales : d'abord déterminer : (a) la représentation des services composites, (b) la population initiale, (c) la fonction d'utilité pour évaluer les compositions et (d) le mécanisme de mise à jour des compositions à la fin de chaque itération de l'algorithme.

   (a) **représentation des services composites :** Cette phase consiste à faire correspondance entre le problème de composition de services et l'algorithme d'optimisation GWO, la position d'un loup représente une solution réalisable, elle est indiquée par un vecteur de dimensions D appelé vecteur ($D - dimensional$) où D est le nombre de tâches abstraites du flux de travail. Chaque élément du vecteur a une valeur indiquant l'indice du service concret sélectionné parmi les services candidats dans un service abstrait.

(b) **Déterminer la population initiale des compositions :** Initialement, n compositions sont choisis au hasard pour chaque tâche abstraite parmi les services candidats dans un ensemble de données.

(c) **Déterminer la fonction d'utilité pour évaluer les compositions :** La fonction de fitness est utilisée pour mesurer la précision du loup correspond à la fonction d'utilité qui permet d'évaluer les services composites sensible à la QoS. Après avoir calculer les valeurs d'utilité de chaque service, les trois meilleurs solutions α ,β ,δ sont déterminées.

(d) **Le mécanisme de mise à jour des compositions :** les compositions dans une population sont modifiées à la fin de chaque itération de l'algorithme GWO en fonction d'*Alpha*, *Beta* et *Delta*, pour se rapprocher de la meilleur solution. Le GWO classique n'est pas approprié pour résoudre des problèmes discrets et comme la composition du service Web a un espace discret (chaque dimension de la composition des services Web est une représentation d'une dimension d'un service concret et ne peut pas accepter des valeurs continues),alors dans cette approche la fonction tangente hyperbolique est utilisée pour convertir les problèmes continus en problèmes discrets.

Discussion et analyse

Dans cette étude, l'utilisation efficace de GWO pour la composition de services Web sensible à la QoS a été étudiée. Pour trouver une composition optimale dans un espace discret indépendamment du nombre d'itérations de l'algorithme. Cependant si un service Web a la meilleure valeur de d'utilité, il sera sélectionné comme solution suggérée ; alors qu'il peut y avoir plusieurs solutions similaires avec des valeurs d'utilité inférieures mais avec des services candidats plus conviviaux . Dans ce cas, ces solutions resteraient à l'écart des utilisateurs.

2. **Mécanisme de composition de services sensible a l'énergie dans les réseaux de capteurs sans fil orienté services**

Dans [Zhao et al., 2017], une approche de composition de services est proposée dans des réseaux de capteurs sans fil orientés services où les contraintes spatiales et temporelles ainsi que l'efficacité énergétique sont prises en compte. Cette approche comprend trois niveaux : a) la construction du réseau de services, b) les recommandations de chaînes de classes de services et c) la composition des services WSN.

(a) **Construction du réseau de services :** les objets intelligents hétérogènes servant de nœuds de capteurs sont encapsulés et représentés comme des services WSN. Un service est un tuple caractérisé par un nom, une description, une opération, l'énergie restante, une contrainte spatiale et une contrainte temporelle. L'opération est caractérisée par un nom, un ensemble de paramètres d'entrée et un ensemble de paramètres de sortie. Les services ayant la même fonctionnalité sont regroupés dans une classe de services, qui est un tuple caractérisé par un nom, une description et une opération. Le calcul de l'invocabilité entre chaque paire de classes de services se fait en considérant : le degré de similitude de leurs noms et descriptions et la possibilité d'invocation de leurs opérations qui est à son tour calculé en considérant : le degré de similitude des paramètres d'entrée de une opération et les paramètres de sortie d'une autre et le degré de similitude de leurs noms, afin d'éviter la classe de services qui a une possibilité d'invocation inférieure à une certaine valeur, pour construire un réseau de services. Ce réseau est représenté par un graphe orienté où les sommets sont les classes de services, les liens directs sont les invocations et le poids spécifié sur les liens reflète la possibilité d'invocation entre les classes de services.

(b) **Recommandations des chaînes de classes de services :** après la construction du réseau de services, la découverte et la recommandation des chaînes de classes de services se font en deux étapes. La première étape consiste à calculer le degré de similitude de la description de l'utilisateur et de la description du service, afin de sélectionner des classes de services candidates qui présentent un degré élevé de similitude. La deuxième étape calcule la possibilité d'invocation pour les paramètres d'entrée (ou paramètres de sortie) spécifiés par l'utilisateur, sur la base des paramètres d'entrée (ou paramètres de sortie) de l'opération de chaque classe de services obtenue à l'étape précédente, de sorte que l'appel le plus élevé Le degré pour une classe de services indiquera l'état de début (ou de fin). Ensuite, un algorithme de recherche de graphe limité en profondeur est utilisé pour sélectionner des classes de services candidates à partir du réseau de services.

(c) **Composition des services WSN :**les chaînes de classes de services obtenues au deuxième niveau sont instanciées dans un certain nombre de services WSN tels que : les contraintes spatiales et temporelles qui doivent être compatibles avec celles requises par une application donnée, la consommation d'énergie incluant l'énergie consommée lors de l'instanciation du WSN services et celle consommée lors de leur communication doivent être faibles, et la consommation d'énergie d'un service candidat WSN doit être équilibrée pour réduire ces derniers et prolonger la durée de

vie du réseau. La sélection des services qui satisfont réellement ces contraintes est un problème d'optimisation multi-objectif multi-contraintes qui est résolu par un algorithme d'optimisation de l'essaim de particules et un algorithme génétique.

L'algorithme d'optimisation de l'essaim partage des compositions optimales globales et locales entre les particules, qui mettent à jour leurs vitesses et leurs positions à chaque itération pour trouver la composition optimale. L'algorithme génétique sélectionne une proportion de services candidats en fonction d'une fonction de fitness pour constituer une nouvelle population. La population suivante est générée en appliquant des croisements et des opérateurs de mutation. Cette procédure est répétée jusqu'à ce qu'une condition d'arrêt soit atteinte, telle qu'une composition approximativement optimale soit trouvée ou qu'un nombre donné de générations ait été atteint.

Discussion et analyse

Ce mécanisme réduit la consommation d'énergie pour promouvoir la composition de services WSN de sorte que la durée de vie du réseau devrait être prolongée ,garantit la satisfaction des contraintes de QOS de l'utilisateur et réduit l'espace de recherche. Néanmoins, le filtrage de services candidats qui ne répondent pas aux contraintes fonctionnelles de l'utilisateur ne réduit pas suffisamment l'espace de recherche de la composition de services qui est fortement influencé par les paramètres d'initialisation.

3. **Composition des grands services sensible a la qualité de service grâce à l'algorithme évolutif basé sur *MapReduce* avec mutation guidée**

Dans [Jatoth et al., 2018] une composition des grands services sensible à la QOS utilisant un algorithme évolutif basé sur $MapReduce$ avec mutation guidée est proposée. L'objectif de l'optimisation serait de trouver X qui maximisera les fonctions d'utilités des attributs de QOS globaux calculés en appliquant récursivement l'agrégation aux éléments consécutifs qui forment la structure de la composition selon la préférence des utilisateurs afin d'obtenir une composition efficace de big services avec une meilleure performance et vitesse de convergence .Cette approche comporte trois phases :

(a) **Phase d'initialisation :** L'opérateur de dominance pareto (Skyline) est utilisé dans le cadre de $MapReduce$ afin de générer la population initiale, ce dernier conserve que les services candidats ayant une utilité maximale dans un ensemble R de services candidats. Ensuite, les services candidats ayant une valeur d'utilité plus proche de

ces services candidats sont également sélectionnés, ce qui réduit l'espace de recherche et filtre les services candidats optimaux pour chaque service abstrait.

Dans cette phase, les services candidats sont stockés sous forme de <clé ;valeur> paires, où la clé représente l'identificateur du service et la valeur représente les informations du service. Ces deux paires sont utilisées comme entrées dans la phase suivante. Les fichiers de service sont stockés sous la forme d'une structure de paires <clé ;valeur> dans un système de fichiers distribué, où la clé est l'ID du service et la valeur est l'information du service. Ce fichier est utilisé comme entrée dans la phase MapReduce.

(b) **Phase MapReduce :** Cette phase est répétée plusieurs fois. Chaque itération représente un travail de *MapReduce*, qui est considéré comme une entrée de l'itération suivante. Un emploi *MapReduce* représente une population actualisée. Cette phase comprend deux fonctions : une fonction $Map()$ et une fonction $Reduce()$. Pour chaque tâche, la fonction $Map()$ divise chaque population en sous-populations qui sont stockée sous la forme $<$ clé ; valeur $>$ paires, comme la clé représente le service abstrait et la valeur la service candidat avec ses attributs de qualité de service. Le résultat de la fonction $Map()$ qui correspond à tous les services candidats et leurs attributs QoS sont envoyés à la fonction $Reduce()$ qui sélectionnera le meilleur service de qualité de service.

(c) **Phase de réparation :** Cette phase est demandée lorsque l'étape précédente fournit des compositions irréalisables. Un opérateur réparateur est appliqué à tous les services candidats de la liste R pour sélectionner un service parmi les services candidats non sélectionnés lors de la phase d'initialisation, une vérification de l'appartenance aux services sélectionnés dans toutes les compositions est effectuée. Dans le cas favorable, il vérifie si la fréquence d'occurrence de ce service est inférieure à un seuil donné ; Notez que la fréquence d'occurrence est calculée en divisant le nombre de compositions dans lesquelles ce service est un service composant sur la taille de la population. Dans le cas où la fréquence d'occurrence est inférieure au seuil donné, le service est ajouté à tous les compositions générées et supprimées de R afin de rendre les compositions réalisables.

Discussion et analyse

Cette approche assure une bonne extensibilité et fournit une composition optimale.Cependant, cette approche ne traite pas de la consommation d'énergie,

sensibilisation aux réseaux et l'hétérogénéité des systèmes.

4. **Composition de service dans QoS-aware cloud à l'aide de la stratégie d'aigle**

Dans [Gavvala et al., 2019],un algorithme de composition de services sensible à la QoS dans cloud computing basé sur la stratégie d'aigle(ESWOA) pour assurer un bon équilibre entre l'exploration et l'exploitation est proposé. Ce dernier comporte quatre phases importantes :

(a) **phase de codage :** Cette phase consiste à faire correspondance entre le problème de composition de services et un algorithme d'optimisation des baleines (WOA), de telle sorte que la position de la baleine est représentée comme une solution réalisable d_x qui est un vecteur de dimension m correspond à une composition de services et la meilleure baleine en termes de fonction de fitness correspond à la meilleure composition en termes de valeur d'utilité.

(b) **La phase d'initialisation :** Cette phase consiste à générer un nombre de compositions SN dans la population de façon aléatoire, ensuite pour chaque composition la valeur d'utilité CSQoS est calculée, la valeur d'utilité la plus élevée représente la meilleur composition.

(c) **La phase d'exploration :** Afin d'assurer la diversification et la recherche globale des compositions. Cette phase consiste à modifier toutes les compositions de la population selon une probabilité *prob*, si la valeur *prob* est inférieure à q (nombre généré aléatoirement). La modification est faite en sélectionnant dans chaque composition un service candidat aléatoirement et en le remplaçant par un autre dans le même service abstrait. La nouvelle composition obtenu remplace l'ancienne lorsque sa valeur d'utilité est supérieur, et si la valeur d'utilité de la nouvelle composition est supérieure à celle de la meilleure composition de la population, alors la nouvelle composition obtenu est désignée comme la meilleure composition de la population. Un paramètre P_e fixé à 0,2 est comparé à un nombre généré aléatoirement. Cette comparaison est utilisée pour décider de la prochaine étape de l'algorithme, si la valeur P_e est inférieure à ce nombre aléatoire, la phase d'exploitation en stratégie d'aigle est exécutée. Sinon, l'algorithme passe à l'autre phase d'exploration.

(d) **La phase d'exploitation :** Au niveau de cette phase les compositions sont améliorées, en utilisant un nombre p généré aléatoirement afin de choisir la meilleure méthode d'amélioration, soit la méthode d'encerclement rétractable dans le cas ou p est supérieur à 0.5, sinon c'est la méthode spirale qui est exécuté.

Discussion et analyse

(ESWOA) est un algorithme qui assure l'équilibre entre la stratégie d'exploration et la stratégie d'exploitation afin de converger vers l'optimum global, car cette équilibre permet de surmonter la convergence prématurée. Cependant Face à des données de grande dimension à grande échelle, il est facile de tomber dans un optimum local et ne peut pas continuer à évoluer.

5. **Optimisation basée sur l'algorithme HHO pour résoudre le problème de composition de service Web sensible à la QoS**

Dans [Li et al., 2020] une approche d'optimisation approximative basée sur l'algorithme HHO pour résoudre le problème de composition de services Web sensible à la QoS dans un espace de recherche discret est proposée . Cette approche comporte cinq phases impotentes :

(a) **phase de codage :** cette phase consiste à faire correspondance entre l'algorithme (HHO) est la composition de services où la position d'un Buse de Harris représente une solution réalisable et celle de la proie comme la meilleure solution de dimension m qui représente le nombre de services abstraits et chaque indice représente le service concret. Vu que QWSC est un problème d'optimisation avec contraintes globale et un problème dur non polynomiale (NP) non déterministe discret [Tang and Xu, 2005] alors il a été modélisé comme un problème de programmation en nombre entier mixte [Zeng et al., 2004], affin d'adopter la méthode de prétraitement pour établir le voisinage continues floues de services concrets(Fuzzy Optimal Continuity Construction (FOCC)) qui rend la stratégie de recherche locale de l'algorithme(CHHO) efficace dans l'espace discret et dans l'espace continue, comme il améliore les performances d'optimisation. La méthode de prétraitement comprend trois étapes importantes :

i. **l'étape de regroupement :** l'algorithme de clustring à moyenne K ($K-mean$) est utilisé pour regrouper les services concrets sous forme de classes (clusters) en fonction de la similitude des valeurs de Qos.

ii. **l'étape de tri des clusters :**les clusters sont triés par ordre croissant selon la valeur SumQos de chaque centre de cluster. Des codages entiers plus grands sont attribués au cluster avec une valeur plus élevée de SumQos.

iii. **Etape de tri des services concrets :**à l'intérieur de chaque cluster ,chaque service concret est trié par sa valeur SumQos dans l'ordre croissant.

(b) **Initialisation :**permet de générer le nombre de compositions aléatoirement dans une population ensuite, la valeur csQos est calculée pour chaque composition la meilleure valeur est identifiée comme la meilleure composition.

(c) **Contrôle des paramètres adaptatifs pour exploitation et exploration :**Le paramètre de contrôle E est utilisé pour ajuster de manière adaptative l'étape d'exploration et d'exploitation de CHHO. Le paramètre E converge progressivement de -2 et 2 vers 0 à fur et à mesure que l'itération augmente.

(d) **L'exploration :**Cette phase est appliquée lorsque le $|E| \geqslant 1$. L'algorithme CHHO met à jour les compositions en utilisant deux stratégies d'exploration différentes. La première stratégie d'exploration stratégique est la co-évolution, la mis à jour d'une composition se fait par rapport à une autre composition générée aléatoirement dans la population. La deuxième stratégie consiste à mettre à jour la composition individuelle actuelle sur la base des informations de la meilleure composition.

(e) **L'exploitation :**L'exploitation se fait lorsque |E| <1, cette étape consiste à diviser la population de la composition en deux sous populations de même nombre, une adopte la stratégie de perturbation unidimensionnelle chaotique logistique (LCSDP). Cette stratégie perturbe d'une façon continue et chaotique une seul dimension de la solution optimale actuelle pendant le processus d'itération, sautant ainsi de l'optimum local et évoluant vers l'optimum globale. L'autre partie de la sous-population adopte les stratégies d'exploitation originales de l'algorithme (HHO) ,si $|E| \geqslant 0.5$ la stratégie du siège doux est utilisée sinon, la stratégie du siège dur .

Analyse et discussion

Cette approche d'optimisation basée sur l'algorithme HHO pour résoudre le problème de composition de services Web sensible à la QoS (QWSC), consiste d'abord à utiliser une méthode de prétraitement des ensembles de données de services concrets appelée FOCC, qui rend la stratégie de recherche locale des algorithmes méta-heuristiques aussi efficace dans l'espace discret que dans l'espace continu, rendant ainsi certains algorithmes méta-heuristiques conçus pour des problèmes continus adaptés à la résolution de QWSC qui est un problème discret. Deuxièmement cette approche combine l'algorithme (HHO) avec une nouvelle stratégie d'exploration LSCDP(engagée à améliorer la capacité de l'algorithme à sortir de l'optimisation locale). Cependant, les performances de l'approche (CHHO) ne sont pas bonnes lorsque des corrélations de QoS existent entre les services [Deng et al., 2014].

2.4 Comparaison et synthèse

Le tableau 2.3 résume les approches de composition de services web sensibles à la qualité de service présentées dans ce chapitre et les compare en termes de : 1) La scalabilité, 2)l'optimalité, 3)sensibilisation à la qualité de service et 4) QoS sensible à la corrélation, tandis que le tableau 2.4 donne les avantages et les limites de chacun d'eux.

2.4.1 La scalabilité

La scalabilité est une métrique qui permet d'évaluer la performance d'un algorithme de composition de services en termes de temps de composition et d'optimalité selon le nombre de services candidats.

2.4.2 l'optimalité

optimalité de la composition permet d'évaluer si la composition obtenue égale à l'optimum ou bien proche de lui.

2.4.3 Sensible à la qualité de service

La sensible à la qualité de service ou (QoS awareness) c'est à dire que l'algorithme de composition de services prenant en compte les contraintes globales de QoS.

2.4.4 QoS sensible à la corrélation

Les attributs de Qos des services web sont corrélés entre eux quand ils sont déployés par le même fournisseur, cela influence sur la qualité globale de la composition des services.

Les approches	**La scalabilité**	**l'optimalité**	**sensible à la qualité de service**	**QoS sensible à la corrélation**
[Deng et al., 2014]	Oui	Optimale	Oui	Oui
[Chen et al., 2015]	Non	Optimale	Oui	Non traité
[Khanouche et al., 2016]	Oui	Quasi optimale	Oui	Non traité
[Kumar and Purohit, 2016]	Non traité	Non évalué	Oui	Non traité
[Wang et al., 2017]	Oui	Non évalué	Oui	Non traité
[Zhao et al., 2017]	Non	Quasi optimale	Non	Non traité
[Karimi and Babamir, 2017]	Oui	Quasi optimale	Oui	Non traité
[Jatoth et al., 2018]	Oui	Optimale	Oui	Non traité
[Purohit and Kumar, 2019]	Oui	Non évalué	Oui	Non traité
[Gavvala et al., 2019]	Oui	Quasi optimale	Oui	Non traité
[Li et al., 2020]	Oui	Quasi optimale	Oui	Non traité
[Khanouche et al., 2020]	Oui	Quasi optimale	Oui	Non traité

TABLE 2.3 – Comparaison des approches de composition de services sensibles à la qualité de service en fonction des défis de composition des services.

Les approches	**Les avantages**	**Les limites**
[Deng et al., 2014]	— Améliorer la QoS des services composites générés en prenant en compte les corrélations de QoS dans le processus de sélection des services. — Réduire l'espace de recherche et le temps d'exécution.	— Cette approche ne considère que les plans de services avec une structure séquentielle.
[Chen et al., 2015]	— Réduire l'espace de recherche et le temps d'exécution.	— La sélection locale ne tient pas compte de la relation entre les services abstraits. — L'efficacité de la composition des services diminue lorsque les services candidats sont augmentés.
[Khanouche et al., 2016]	— Réduire la consommation d'énergie des services candidats. — Réduire le temps de sélection sans affecter la satisfaction du consommateur.	— La sélection locale ne tient pas compte des contraintes de QoS globales.
[Kumar and Purohit, 2016]	— Réduire l'espace de recherche.	— L'algorithme n'est pas fiable lors de l'augmentation des services web candidats .
[Wang et al., 2017]	— Réduire le temps de recherche. — Assure la fiabilité et prendre en considération la fluctuation de la QoS.	— Ne supporte pas de nouveaux candidats ou bien ceux rarement utilisés. — Le calcule de fluctuation de QoS ne peut être effectué pour le prix.
[Zhao et al., 2017],	— Réduire la consommation d'énergie et l'espace de recherche. — Garantir la satisfaction des contraintes de QOS de l'utilisateur.	— La composition des services est fortement influencée par les paramètres d'initialisation. — Le filtrage des services candidats qui ne répondent pas aux contraintes fonctionnelles de l'utilisateur ne réduit pas suffisamment l'espace de recherche de la composition.

[Karimi and Babamir, 2017]	— trouver une composition quasi optimale dans un espace discret.	— Cette approche à une faible capacité d'exploration. — Ne prend pas en considération les contraintes locales.
[Jatoth et al., 2018]	— Assure une bonne extensibilité. — Fournir une composition optimale.	— Ne traite pas la consommation d'énergie.
[Purohit and Kumar, 2019]	— Réduire l'espace de recherche.	— Le calcul du skyline prend beaucoup de temps.
[Gavvala et al., 2019]	— assurer l'équilibre entre l'exploration et l'exploitation, afin d'éviter la convergence rapide et de trouver l'optimum globale.	— Ne prend pas en compte des interdépendances et des corrélations entre les services et les attributs de la QoS.
[Li et al., 2020]	— Le prétraitement pour les ensembles de données de services concrets (FOCC),rend la stratégie de recherche locale de l'algorithme proposé aussi efficace dans un espace discret que dans un espace continu. — La nouvelle stratégie LSCDP d'exploration permet à l'algorithme (CHHO) de sortir de l'optimisation locale.	— Lorsque des corrélations de QoS existent entre les services cela conduit à la baisse des performances de l'algorithme.
[Khanouche et al., 2020]	— Rendre les contraintes de qualité de services de l'utilisateur plus flexibles afin d'augmenter la faisabilité de la composition. — Réduire l'espace de recherche de la composition et diminuer le temps de composition. — Assurer une valeur d'optimalité élevée	— cette approche ne prend pas en considération les environnements dynamiques.

TABLE 2.4 – Comparaison des approches de composition de services sensibles à la qualité de service en termes d'avantages et de limites.

2.5 Discussion

Les travaux étudiées montrent que le problème de composition des services sensible à la qualité de service a été largement abordé dans la littérature. La majorité des ap-

proches stochastiques de composition de services [Zhao et al., 2017, Karimi and Babamir, 2017, Gavvala et al., 2019, Li et al., 2020] basées sur les algorithmes méta-heuristique, permettent d'explorer l'espace de recherche (souvent très grand) efficacement d'une façon aléatoire afin de déterminer une composition quasi optimal en termes de QoS, en un temps d'exécution raisonnable. Les approches méta-heuristique génèrent une population initiale aléatoire ce qui conduit à un temps de convergence élevé et surtout pour les approche qui assure l'équilibre entre la stratégie d'exploitation et la stratégie d'exploration tels que[Gavvala et al., 2019, Li et al., 2020], et pour les approches qui ont une faible capacité d'exploration souffrent du problème de la convergence prématuré. Les approches basées sur l'optimalité de pareto et clustering sont des approches déterministes [Chen et al., 2015, Kumar and Purohit, 2016, Wang et al., 2017, Purohit and Kumar, 2019] qui permettent de comparer toutes les composition dans l'espace de recherche et filtrer les services dominés par d'autres pour obtenir des compositions optimal mais, l'augmentation exponentielle des compositions avec le nombre de services provoque un temps de composition élevé.

Dans ce mémoire, nous abordons le défi de la composition des services sensible à la qualité de service en utilisant l'algorithme des feux d'artifice(FWA).

La méthode FWA est un algorithme qui permet de mettre à jour la population à chaque itération, cela assure une grande capacité d'exploration dans l'espace de recherche et permet d'atteindre l'optimum globale dans un temps raisonnable. Contrairement à d'autres approches basées sur la population proposés dans la littérature, qui ont une faible capacité d'exploration coincent facilement dans l'optimum locale et ne peuvent pas continuer à évoluer. De plus dans le cas où une approche assure l'équilibre entre l'exploration et l'exploitation, dans le bute d'éviter de tomber dans l'optimum locale, le passage d'une stratégie à une autre nécessite plus de temps pour trouver la composition la plus proche de l'optimum.

2.6 conclusion

Nous avons vu dans le présent chapitre quelques approches de composition de services avec QoS à savoir : Les approches basées sur la dominance de pareto, celles basées sur la méta-heuristique pour la composition du service sensible à la QoS, celles basées sur le clustering et sur la QoS pour les applications orientées services Et enfin celles basées sur la corrélation en les classant en deux catégories. Toutefois, ces approches souffrent de limitations comme la possibilité d'une convergence prématuré ou de tomber dans un optimum locale [Birogul, 2019].

Pour résoudre ce problème, nous avons proposé une approche de composition de services basées sur la population dans le chapitre suivant.

Chapitre 3

Approche de composition de services sensible à la QoS basée sur l'algorithme des feux d'artifice FWA

3.1 Introduction

Dans le chapitre précédent, nous avons étudié en détail la problématique de composition et de sélection de services ainsi que les différentes solutions proposées dans la littérature afin de traiter ce problème.

Le défi de la composition de services consiste à trouver la composition qui offre la meilleure valeur d'utilité en termes de QoS et qui satisfait les contraintes globales de l'utilisateur.

Dans ce chapitre, nous présontons notre solution au problème de sélection et de composition de services, pour cela nous proposons une approche basée sur une méta-heuristique afin d'obtenir optimum globale et de ne pas tomber dans optimum local.
Notre approche utilise l'algorithme des feux d'artifices et elle prend en compte les préférences de l'utilisateur sur les paramètres de QoS. Notre solution comprend trois phases : la première initialise les paramètre de l'algorithme, la seconde nous permettra de générer plus de services composites dans l'espace de recherche, la dernière phase est la sélection des services offrant les meilleures compositions.

3.2 L'algorithme des feux d'artifice

inspiré de l'explosion des feux d'artifice dans le ciel nocturne et illuminent les environs[Tan and Zhu, 2010] ont proposés l'algorithme de feux d'artifice(FWA). Dans FWA, un feu d'artifice est considéré comme une solution viable dans l'espace de recherche du problème d'optimisation. De plus, le processus d'allumage des feux d'artifices est considéré comme le processus de fouille du voisinage de ces feux d'artifice, aussi cette algorithme a une forte capacité d'exploration ce qui permet de s'approcher le l'optimum globale.

Dans l'algorithme FWA, pour chaque génération d'explosion, nous sélectionnons d'abord les emplacements, où SP feux d'artifices sont déclenchés. Après l'explosion, les emplacements des étincelles sont obtenus et évalués. Lorsque l'emplacement optimal est trouvé, l'algorithme s'arrête. Sinon, d'autres emplacements sont sélectionnés pour la prochaine génération d'explosion.

Les étapes spécifiques de l'algorithme des feux d'artifices sont détaillées dans les sous-sections suivantes.

3.2.1 Initialisation

Dans cette phase tous les paramètres à utiliser dans l'algorithme sont initialisés tels que la taille de la population SP et le nombre maximale d'itérations $MaxIter$. La population initiale des feux d'artifices Pop est générée aléatoirement. Les feux d'artifice ainsi que les étincelles représentent les solutions réalisables au problème d'optimisation.

La valeur d'utilité de chaque feu d'artifice est calculée et sur la base de cette valeur, le meilleur feu d'artifice x^* de la population est identifié.

3.2.2 Générer des étincelles et déterminer leurs emplacements

Cette étape consiste à :

— A calculer S_i le nombre d'étincelles $\overline{x}_j$ générées par chaque feu d'artifice x_i et à calculer l'amplitude d'explosion A_i de chaque feu d'artifice comme indiqué dans les équations (3.1),(3.2).

$$S_i = m \cdot \frac{y_{max} - f(x_i) + \xi}{\sum_{i=1}^{SP} (y_{max} - f(x_i)) + \xi} \tag{3.1}$$

$$A_i = \Psi \cdot \frac{f(x_i) - y_{min} + \xi}{\sum_{i=1}^{SP} (f(x_i - y_{min}) + \xi} \tag{3.2}$$

où $y_{min} = min(f(x_i)), (i = 1, 2, ..., SP)$ est la valeur de fitness minimale parmi les SP feux d'artifices. $y_{max} = max(f(x_i)), (i = 1, 2, ..., SP)$ est la valeur de fitness maximale parmi les SP feux d'artifices. m et Ψ sont des paramètres contrôlant le nombre total d'étincelles et l'amplitude maximale de l'explosion respectivement. ξ est la plus petite constante de l'ordinateur, utilisée pour éviter une erreur de division nulle.

Pour éviter que les bons feux d'artifices produisent des étincelles beaucoup plus explosives que les feux d'artifices de mauvaises qualités, l'équation(3.3) est utilisée pour limiter le nombre d'étincelles générées :

$$S_i = \begin{cases} round\,(h \cdot m) & \text{si } S_i < hm \\ round\,(b \cdot m) & \text{si } S_i > bm, h < b < 1 \\ round\,(S_i) & \text{sinon} \end{cases} \tag{3.3}$$

Où h et b sont des constantes, et $round()$ représente la fonction prédéfinie rounding.

— trouver les emplacements des étincelles $\overline{x}_j$, en sélectionnant d'abord k dimensions au hasard qui doit être modifier selon l'équation(3.4), ensuite calculer le déplacement de chaque étincelle selon l'équation(3.5) et au finale l'emplacement de chaque étincelle générée est calculé selon l'équation(3.6).

$$Z_k = round\,(rand\,(0, 1))\,, k = (1, 2...., d) \tag{3.4}$$

$$H = A_i \cdot rand\,(-1, 1) \tag{3.5}$$

$$\overline{x}_k^j = \overline{x}_k^j + H \tag{3.6}$$

où d est la dimension de l'emplacement de l'étincelle, et $rand()$ est une fonction qui donne des résultats aléatoire.

Pour conserver la diversité des étincelles, une autre méthode appelé l'explosion gaussienne, est utilisée pour générer des étincelles. Cette dernière consiste à sélectionner au hasard l feux d'artifices. Ensuite pour chaque feu d'artifice sélectionné un certain nombre k de dimensions sont sélectionnées aléatoirement et modifiées, comme indiqué dans les équations (3.4) et(3.7) respectivement.

$$\overline{x}_k^j = \overline{x}_k^j \cdot Gaussian\,(1,1) \tag{3.7}$$

où $Gaussian(1,1)$ est une fonction, qui désigne une distribution gaussienne avec moyenne 1 et écart type 1, elle est utilisé pour définir le coefficient de l'explosion.

3.2.3 Sélection d'emplacements

Dans cette phase un nombre d'emplacements sont sélectionnés afin de générer une nouvelle population. Dans FWA,le meilleur emplacement x^* en fonction de la valeur de fitness est toujours conservé pour la prochaine itération. Ensuite, $SP-1$ emplacements des deux types d'étincelles générées et ceux des feux d'artifices de la population initiale sont sélectionnés en fonction de leurs distance par rapport aux autres emplacements. La distance de ces derniers(emplacements) est défini selon l'équation (3.8).

$$R(x_i) = \sum_{j \in K} d\,(x_i, x_j) \tag{3.8}$$

où K est l'ensemble de tous les emplacements actuels, y compris les feux d'artifice de la population initiale et les deux types d'étincelles générées(sans le meilleur emplacement). Ensuite, la probabilité de sélection d'un emplacement x_i est définie comme suit :

$$P(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)} \tag{3.9}$$

Enfin, les emplacements avec les valeurs de probabilité les plus élevées sont sélectionnés.

La figure(3.1) montre les différentes étapes de l'algorithme FWA.

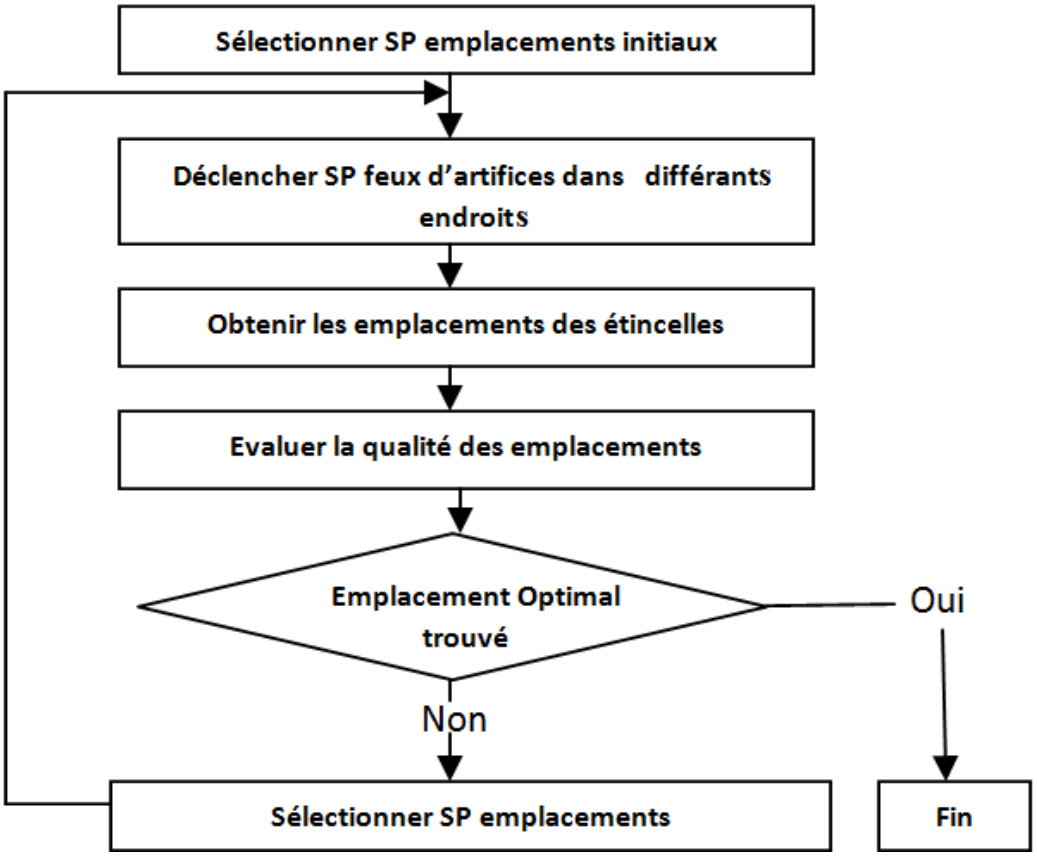


FIGURE 3.1 – Algorithme de feux d'artifice.
Source : [Tan and Zhu, 2010].

A partir de la figure(3.1), on peut voir que le succès de FWA réside dans une bonne conception du processus d'explosion et une méthode appropriée pour sélectionner les emplacements.

Le pseudo code de l'algorithme FWA est présenté dans l'algorithme.

Algorithm 1 PSEUDO CODE DE FWA

Entrée(s) $MaxIter$,Pop et SP
1: Générer aléatoirement la population Pop de taille SP
2: Initialiser m, h, b, Ψ, H et l
3: **Tant que** critères d'arrêt = faux **Faire**
4: Déclenchez respectivement SP feux d'artifice aux SP emplacements
5: **Pour** chaque feux d'artifice x_i **Faire**
6: Calculer le nombre d'étincelle à générer et l'amplitude des feux d'artifice, (3.1),(3.2), (3.3)
7: obtenir les emplacement des étincelles
8: **Fin du pour**
9: **Pour** j=1 à l **Faire**
10: Sélectionner aléatoirement un feu d'artifice
11: Générez une étincelle spécifique pour le feu d'artifice en utilisant l'équation (3.7)
12: **Fin du pour**
13: Sélectionnez le meilleur emplacement x^* et conservez-le pour la prochaine génération d'explosion
14: Sélectionnez aléatoirement $SP-1$ emplacements parmi les deux types d'étincelles et les feux d'artifice actuels selon les équations (3.8), (3.9)
15: **Fin du tant que**
Sortie(s) x^*

3.3 La composition de services web basée sur l'algorithme des feux d'artifices

Dans cette étude, le problème de composition de services sensible à la QoS (QoSaware services composition) est modélisé et réalisé a l'aide de FWA. Un ensemble des feux d'artifice dans FWA représente la population de la composition, un feu d'artifice/étincelle $X_l = \left(x^1_{l,j}, .., x^i_{l,j}, .., x^D_{l,j}\right)$ représente la $l^{\text{ème}}$composition $SC_l = \left(x^1_{l,j}, .., x^i_{l,j}, .., x^m_{l,j}\right)$ où $x^i_{l,j}$ est l'indice du $j^{\text{ème}}$ service candidat dans le $i^{\text{ème}}$ service abstrait de la $l^{\text{ème}}$ composition. SC^* représente la meilleure composition avec la valeur d'utilité la plus élevée en termes de QoS. Le tableau (3.1) montre la correspondance entre les termes utilisés dans l'algorithme FWA et le problème de composition de services :

L'algorithme de FWA	**Le problème de composition de services.**
Ensemble des feux d'artifices	La population contenant la composition.
Une étincelle	Service composite dans la population.
Dimension	Le nombre de services abstraits.
Emplacement d'une étincelle	L'indice d'un service candidat dans une composition.
La fonction de fitness	La valeur d'utilité en termes de QoS.
La meilleure étincelle x^*	Le meilleur service composite en termes de valeur d'utilité.

TABLE 3.1 – la correspondance entre les termes utilisés dans l'algorithme FWA et le problème de composition de services.

Dans notre solution nous avons adopté un shéma de codage entier similaire à[Huo et al., 2015] comme illustré dans la figure 3.2 qui permet de lier un entier au service concret :

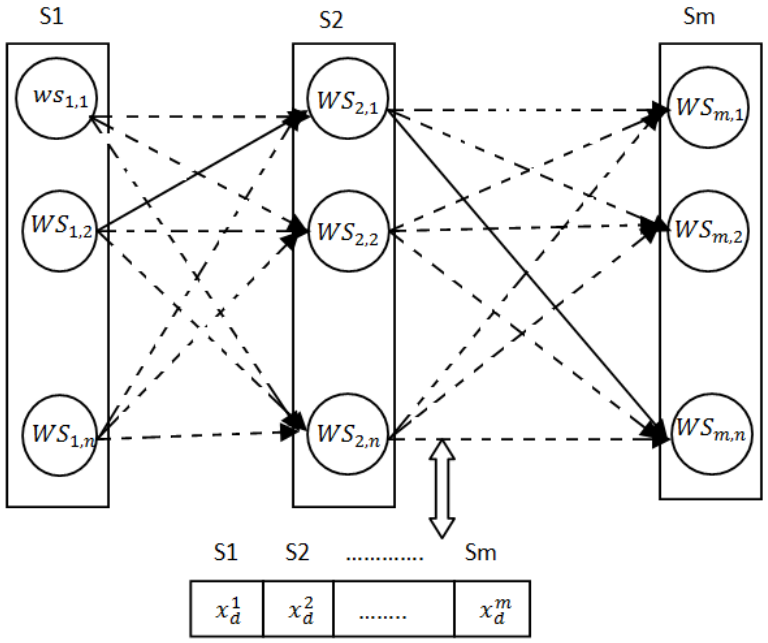


FIGURE 3.2 – Schéma de codage d'un tableau d'entiers.
Source : [Huo et al., 2015].

Les positions des étincelles et les feux d'artifices X_l sont représentées comme un vecteur de D dimensions $X_l = \left(x_{l,j}^1, .., x_{l,j}^i, .., x_{l,j}^D\right)$,dans le tableau de nombre entier chaque élément $x_{l,j}^i$ représente la valeur du $j^{\text{ème}}$ service concret candidat WS du $i^{\text{ème}}$ service abstrait, cette valeur est compris dans l'intervalle $[lb, ub]$ où la valeur de lb est à 1 et ub est le nombre de services concrets candidats pour chaque service abstrait.

Dans ce qui suit, nous présentons les différentes phases de notre algorithme.

La phase d'initialisation : Dans cette étape les paramètres de l'algorithme sont initialisés tels que le nombre maximale d'itérations $Maxiter$, le nombre de compositions dans la population SP. La population initiale Pop des compositions est générée aléatoirement [Huo et al., 2015] en utilisant l'équation suivante :

$$WS_{l,j}^{i} = lb + round\left(rand\left(0,1\right) * \left(ub - lb\right)\right) \tag{3.10}$$

La valeur d'utilité de chaque composition dans la population Pop est calculée ensuite, la meilleure composition SC^* en termes de valeur d'utilité est identifiée, la valeur d'utilité est calculée selon l'équation(2.1). L'algorithme(2) résume l'étape d'initialisation.

Algorithm 2 INITIALISATION

Entrée(s) les services candidats

1: Initialiser $Maxiter$ et SP
2: Générer une population Pop de composition SP a partir des services candidats en utilisant l'équation(3.10)
3: **Pour** chaque composition dans Pop **Faire**
4: Calculer $F(SC)$
5: **Fin du pour**
6: **Pour** chaque valeur d'utilité F **Faire**
7: Déterminer la meilleure composition SC^* dans la population
8: **Fin du pour**

Sortie(s) la population initiale

La phase de génération des services composites : Cette étape nous permet de générer plus de services composites dans l'espace de recherche, en calculant d'abord pour chaque service composite initial dans la population Pop, la valeur S_{Act} le nombre de services composites à générer et la valeur A_{Act} issu de l'amplitude d'un feu d'artifice, selon les équations suivantes :

$$S_{Act} = m \cdot \frac{F(SC^*) - F(SC_{Act}) + \xi}{\sum_{Act=1}^{sp}\left(F(SC^*) - F(SC_{Act})\right)) + \xi} \tag{3.11}$$

$$A_{Act} = \Psi \cdot \frac{F(SC_{Act}) - F(SC_W) + \xi}{\sum_{Act=1}^{sp}\left(F(SC_{Act}) - F(SC_W)\right) + \xi} \tag{3.12}$$

Où SP représente le nombre de services composites dans la population, SC_W est le service composite avec la valeur d'utilité minimale, ξ c'est la plus petite constante dans l'ordinateur pour éviter la division par zéro m et Ψ sont des constantes pour contrôler le nombre de services et l'amplitude respectivement. Afin de limiter le nombre de services composites à générer on définie S_{Act} comme suit :

$$S_{Act} = \begin{cases} round\,(h \cdot m) & \text{si } S_{Act} < hm \\ round\,(b \cdot m) & \text{si } S_{Act} > bm, h < b < 1 \\ round\,(S_{Act}) & \text{sinon} \end{cases} \tag{3.13}$$

où h et b sont des constantes. l'équation(3.14) permet de mettre à jour le service candidat d'une composition.

$$WS^K_{Act,j} = WS^k_{Act,j} + round\,(A_{Act} \cdot rand\,(-1,1)) \tag{3.14}$$

Si la valeur de $WS^K_{Act,j}$ est au-delà de la limite à la dimension k l'équation(3.15) permet de passer vers une nouvelle position .

$$WS^K_{Act,j} = WS^k_{Act,min} + |WS^K_{Act,j}| mod\left(WS^K_{Act,max} - WS^K_{Act,min}\right) \tag{3.15}$$

Où $WS^K_{Act,max}$ et $WS^K_{Act,min}$ sont les positions maximale et minimale respectivement à la dimension K d'un service candidat d'une composition SC_{Act}.
L'algorithme(3) résume la phase de génération des services composites.

Algorithm 3 Génération des services composites

Entrée(s) Le service composite SC_{Act}

1: Initialiser $SC_{New_j} = SC_{Act}$, $W=\phi$.
2: Sélectionner aléatoirement un ensemble de dimensions Z_K de services composites SC_{Act} où les services web WS vont être modifiés équation(3.4).
3: **Pour** chaque $WS^K_{Act,j}$ de SC_{New_j} **Faire**
4: **Si** $Z_K == 1$ **Alors**
5: appeler l'équation(3.14).
6: **Si** $WS^K_{Act,j} < WS^K_{Act,min}$ ou $WS^K_{Act,j} > WS^K_{Act,max}$ **Alors**
7: appeler l'équation(3.15).
8: **Fin du si**
9: **Fin du si**
10: **Fin du pour**
11: $W = W \cup SC_{New_j}$

Sortie(s) W

Afin d'augmenter la diversité de la population, l'algorithme des feux d'artifices à introduit un opérateur de mutation selon équation(3.16), pour générer d'autres services composite issu de l'opérateur de mutation .

$$WS^K_{Act,j} = WS^K_{Act,j}.Gaussian\,(1,1) \tag{3.16}$$

L'algirithme(4) permet de générer d'autres services composites spécifiques.

Algorithm 4 Génération des services composites spécifiques

Entrée(s) Le service composite SC_{Act}

1: Initialiser $SC_{New_j} = SC_{Act}$, $W=\phi$.
2: Sélectionner aléatoirement un ensemble de dimensions Z_K de services composites SC_{Act} où les services web WS vont être modifiés équation(3.4).
3: **Pour** chaque $WS^K_{Act,j}$ de SC_{New_j} **Faire**
4: **Si** $Z_K == 1$ **Alors**
5: appeler l'équation(3.16).
6: **Si** $WS^K_{Act,j} < WS^K_{Act,min}$ ou $WS^K_{Act,j} > WS^K_{Act,max}$ **Alors**
7: appeler l'équation(3.15).
8: **Fin du si**
9: **Fin du si**
10: **Fin du pour**
11: $W = W \cup SC_{New_j}$

Sortie(s) W

La phase de sélection des services composites : Cette phase consiste a générer une nouvelle population de services composites. En effet cette phase comporte plusieurs étapes ; d'abord combiner les services composites générés dans la phase de génération de services composites plus les services de la population Pop dans un ensemble K, le meilleur service SC^* en fonction de la valeur d'utilité est toujours conservé en premier pour la prochaine itération, après cela SP–1 services composites SC sont sélectionnés en fonction de leurs distances par rapport aux autres SC comme le montre l'équation suivante :

$$R(SC_i) = \sum_{j \in K} d\left(SC_i, SC_j\right) \tag{3.17}$$

Ensuite, la probabilité de sélection de chaque service composite est calculé selon l'équation suivante :

$$P(SC_i) = \frac{R(SC_i)}{\sum_{j \in K} R(SC_j)} \tag{3.18}$$

Enfin $SP-1$ services composites avec les valeurs de probabilité les plus élevés sont sélectionnés pour la prochaine itération ,y compris SC^* afin d'obtenir une nouvelle population Pop de taille SP.

Les phases citées sont répétées plusieurs fois jusqu'à ce que le critère d'arrêt égale à vrai, lorsque le nombre d'itérations maximale est atteint.

Le pseudo-code de l'algorithme FW-WSC est décrit par l'algorithme(5).

Algorithm 5 PSEUDO-CODE FW-WSC

Entrée(s) *Maxiter*, *SP*, *Pop*.

Réalisation de la phase d'initialisation (Algorithme 2).
Initialiser ξ,m,h,b ,l et Ψ.
Tant que critères d'arrêt = faux **Faire**
  Pour $i = 1$ jusqu'à SP **Faire**
    Calculer S_{Act} et A_{Act} selon les équations (3.11) ,(3.12) et (3.13)
    Pour $j = 1$ jusqu'à S_{Act} **Faire**
      Générer un ensemble W_1 de services composites (Algorithme 3)
    Fin du pour
  Fin du pour
  Sélectionner aléatoirement l services composites
  Pour $j = 1$ à l **Faire**
    Générer un ensemble W_2 de services composites (Algorithme 4)
  Fin du pour
  $K = W_1 \cup W_2 \cup Pop$
  Sélectionner le meilleur service $SC^* \in K$
  Pour chaque $SC_i \in (K - SC^*)$ **Faire**
    Calculer la distance entre le service SC_i et les autres. Équation (3.17)
  Fin du pour
  Pour chaque $SC_i \in (K - SC^*)$ **Faire**
    Calculer la probabilité de sélection de service SC_i.Équation (3.18)
  Fin du pour
  Mettre les probabilités dans l'ordre décroissant et sélectionner (SP–1) services composites selon l'ordre des probabilités.
  Mettre les services sélectionnés ainsi SC^* dans *Pop*.
Fin du tant que

Sortie(s) meilleur service composite SC^*

3.4 Conclusion

Dans ce chapitre, nous avons présentés notre approche de composition de services sensible à la QoS basée sur un algorithme méta-heuristique des feux d'artifices qui prend en compte plusieurs facteurs, à savoir ; la fiabilité des compositions ainsi que les préférences de l'utilisateur. Afin de valider l'approche proposé, le chapitre suivant sera consacré à la simulation et à l'évaluation des performances de l'algorithme proposé.

Chapitre 4

Simulation et analyse des performances

4.1 Introduction

Dans ce chapitre, nous décrivons l'évaluation des performances de l'algorithme (FW-WSC) de composition de services sensible à la qualité de service proposé selon un scénario de simulation. Tout d'abord, nous présentons l'environnement de simulation. Ensuite, nous décrivons le scénario et l'ensemble des données utilisées dans l'évaluation ainsi que les mesures de performance considérées. Enfin, nous présentons les résultats obtenus de l'algorithme proposé et sa comparaison avec d'autres algorithmes proposés dans la litérature, qui sont[Karimi and Babamir, 2017, Gavvala et al., 2019].

4.2 Scénario et méthodologie

4.2.1 Environnement de simulation et jeu de données

L'algorithme de composition de services sensible à la QoS proposé a été implémenté et évalué à l'aide de Matlab R2018b fonctionnant sur un PC équipé d'un système Windows 64 bits et d'un Intel Celeron CPU N2929 avec une fréquence de 1.86 GHz et 4 Go de RAM. Les compositions de services considérées dans le scénario de simulation ont une structure séquentielle.

Nous avons choisi comme environnement de simulation MATLAB pour les raisons suivantes :

1. C'est un langage de programmation de calcul scientifique et de visualisation graphique, développé par la société The MathWorks. Il est disponible sur plusieurs plateformes.
2. Il dispose d'une large bibliothèque de fonctions prédéfinies avec des notations simples et puissantes, donc optimiser le code des programmes.

3. Il permet de manipuler des matrices, d'afficher des courbes des données et de mettre en ouvre des algorithmes etc...

4.2.2 références de comparaison

La performance de l'algorithme FW-WSC est comparé à celles des approches suivantes :

- L'approche de composition de services web sensible a la Qos basé sur l'algorithme des loups gris (GWO)[Karimi and Babamir, 2017] cette approche a été choisie pour la comparaison avec notre approche proposé car cette antécédente à une forte capacité d'exploitation et une faible capacité d'exploration [Yue et al., 2020] contrairement à la notre, afin de voire l'impacte d'exploration sur le type d'optimum trouvé (local/globale).

- L'algorithme de composition des services basé sur la stratégie d'aigle avec l'optimisation des baleines (ESWOA) [Gavvala et al., 2019], cette approche basé sur la population assure l'équilibre entre la stratégie d'exploitation et l'exploration afin, d'éviter l'optimum locale.

4.2.3 Métriques de performances

Pour démontrer l'efficacité de l'algorithme FW-WSC par rapport aux approches susmentionnées, les métriques suivantes sont utilisées dans cette étude :

Temps de composition :représente le temps d'exécution d'un algorithme de composition de services nécessaire pour trouver la composition optimal ou quasi-optimale en termes de QoS.

Valeur d'utilité : cette métrique représente la somme pondérée des valeurs d'attributs de QoS calculée en utilisons une fonction définit, cette valeur permet de trouver la meilleure composition en termes de Qos obtenu par un algorithme de composition.

4.2.4 Paramètres de simulation

L'évaluation de performance de l'algorithme FW-WSC est réalisée en tenant compte des paramètres décrits dans le tableau (4.1).

Paramètre	**Valeur**
Le nombre maximum d'itérations.	200
La taille de la population.	40
Nombre de services concret.	de 2000 à 10000 .
Nombre de services abstraits	5.
Nombre de simulation	10.
Nombre de paramètres de QoS	5.

TABLE 4.1 – Paramètres de simulation.

Le scénario de composition de services est réalisé en simulation pour évaluer la performance de algorithme FW-WSC. Le scénario vise a évaluer l'impact du nombre de services candidats sur les valeurs d'utilités, ainsi le temps de composition. Dans ce scénario le nombre maximal d'itération et fixé a 200 , le nombre de service abstrait à 5, le nombre de services concrets pour chaque service abstrait varie de 2000 à 10000 services et le nombre de paramètre de Qos est fixé à 5. Les services sont issus du dataset QWS[Al-Masri and Mahmoud, 2008]

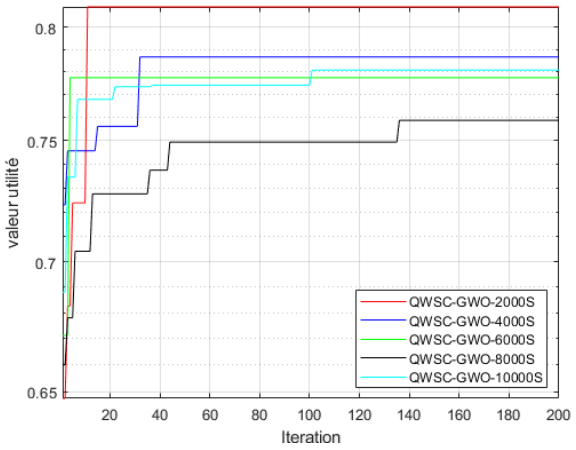
4.3 Performances de l'algorithme FW-WSC

4.3.1 Valeurs d'utilité vs le nombre d'itération

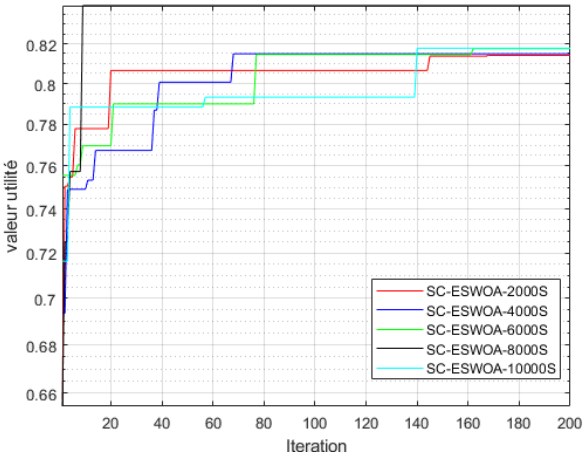
Dans cette simulation, nous mesurons les valeurs d'utilités en termes de QoS obtenues avec les algorithmes FW-WSC, QWSC-GWO et SC-ESWOA en fonction du nombre d'itérations. On observe sur les figures 4.1(a), 4.1(b) et 4.1(c) que les valeurs d'utilités obtenues avec les trois algorithmes augmentent avec le nombre d'itérations jusqu'à atteindre la valeur la plus élevée et restent stables. De plus les résultats e simulation montre que notre algorithme atteint la valeur d'utilité maximale avec un nombre d'itération plus réduit que les autres algorithmes. Pour tous les cas de simulation notre algorithme retourne la meilleure utilité ou bout de la 40 ème itération alors que les autres vont jusqu'à la 140 ème itération.

Les figures (4.1) et (4.2) montrent que le nombre de services candidats a un impact sur la valeur d'utilité dans toutes les approches. FW-WSC donne la meilleure valeur d'utilité quel que soit le nombre de services concrets candidats par rapport aux autres algorithmes, grâce à sa forte capacité d'exploration et sa capacité de générer un nouvel espace de recherche pour chaque itération, ce qui rend la population plus diversifiée par nature. L'espace de recherche devient plus large avec l'augmentation des valeurs de ψ, m et le nombre d'itérations. Cette caractéristique augmente la probabilité d'atteindre l'optimum global, par contre pour l'algorithme QWSC-

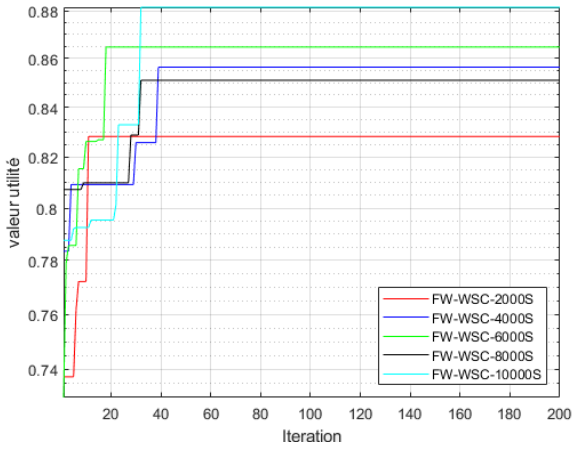
GWO à une faible capacité d'exploration, alors il converge rapidement.



(a) Algorithme(QWSC-GWO)



(b) Algorithme(SC-ESWOA)



(c) Algorithme(FW-WSC)

FIGURE 4.1 – Valeurs d'utilité vs le nombre d'itération.

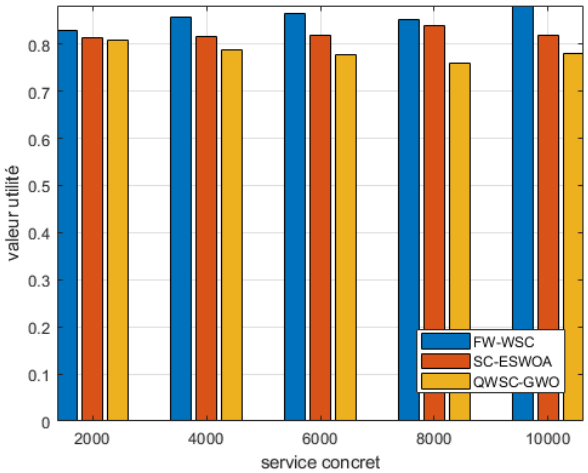


FIGURE 4.2 – La comparaison entre les trois algorithmes selon le nombre de service concret.

4.3.2 Temps de composition vs nombre de services candidats

Dans cette simulation, les algorithmes FW-WSC , QWSC-GWO et SC-ESWOA sont comparés en termes de temps de composition en faisant varier le nombre de services candidats. Le tableau (4.2) montrent que le temps de composition de l'algorithme QWSC-GWO augmente considérablement avec le nombre de services candidats, car chaque composition est évaluée pour chaque itération, ce qui prend plus de temps pour l'exécution.

SC-ESWOA et FW-WSC donnent un bon résultat en termes de valeur d'utilité vu qu'ils ont une grande capacité d'exploration mais concernant le temps de composition SC-ESWOA est beaucoup plus élevé, car une nouvelle population est générée pour chaque itération et chaque composition est évaluée ce qui entraîne un taux de convergence lent. Par contre le temps de composition de FW-WSC est considérablement faible, cela est dû à son temps de convergence en plus de sa capacité à trouver la meilleure composition au bout d'un nombre d'itérations réduit.

Algorithme	Nombre de services candidats	Temps de composition(ms)	Valeur d'utilité
FW-WSC	**2000**	**1.0212**	**0.82531**
	4000	**0.17024**	**0.87556**
	6000	**0.20541**	**0.86202**
	8000	**0.14657**	**0.84768**
	10000	**0.12896**	**0.82889**
QWSC-GWO	**2000**	**4.7583**	**0.78225**
	4000	**7.5173**	**0.79282**
	6000	**7.8882**	**0.80471**
	8000	**7.9911**	**0.79929**
	10000	**9.1485**	**0.7844**
SC-ESWOA	**2000**	**37.2854**	**0.83791**
	4000	**49.2453**	**0.79867**
	6000	**47.6451**	**0.82351**
	8000	**49.214**	**0.82123**
	10000	**53.2108**	**0.82437**

TABLE 4.2 – Comparaison entre FW-WSC, QWSC-GWO et SC-ESWOA en terme de temps de composition.

4.4 Conclusion

Au cours de ce chapitre, nous avons réalisé une évaluation de performance de la solution proposé de la composition de service web sensible à la Qos basé sur l'algorithme de feux d'artifice FW-WSC. Les résultats montrent que l'approche proposée réduit considérablement le temps de la composition, cela est dû au faite que FW-WSC nécessite moins d'itération avant de retrouver la meilleure composition grâce à sa puissance d'exploration. Vu que à chaque itération la taille de l'espace de recherche augmente cela assure la diversité des compositions, ce qui permet de trouver l'optimum globale.

Conclusion Générale

A cause de l'augmentation des services web sur internet, il est difficile de découvrir un service qui répond aux besoins de l'utilisateur. Pour cela plusieurs services peuvent interagir et échanger dynamiquement des informations. La composition de services vise la construction automatique de services complexes à base de services atomiques, tout en respectant les contraintes de QoS imposées par l'utilisateur. En effet l'augmentation exponentielle des compositions avec le nombre de service provoque un temps de composition très élevé avant de pouvoir trouver la meilleure composition. Plusieurs font ressources à des solutions méta-heuristique permettant de trouver des solutions quasi optimale dans un temps raisonnable, cependant la plus part des solutions soufrent de problème de convergence rapide et leurs faible capacité d'exploration.

Dans ce travail nous avant proposé une solution de sélection et de composition de services basé sur l'algorithme méta-heuristique des feux d'artifices (FWA). Le principe de cette dernière consiste de trouver une solution optimale globale, dans un temps raisonnable et de tenir en compte le critère de passage à l'échelle tout en prenant en compte les contraintes des utilisateurs sur les paramètres de QoS. Dans cette approche, le problème de composition de services a été modélisé selon le comportement des feux d'artifices.

L'approche consiste tout d'abord, à générer des services composites a partir de la population initiale pour assurer la diversité, ensuite la probabilité de sélection est calculer pour chaque service, après une sélection de services est faite selon l'ordre décroissent des probabilités afin de former une nouvelle population, et de déterminer la composition la plus proche de l'optimum, cette procédure est répétée plusieurs fois selon un nombre d'itération fixé à l'avance. L'avantage de l'approche proposée réside dans le fait qu'elle nécessite un nombre d'itération réduit avant de trouver la meilleure solution tout en préservant une forte exploration. Les résultats de la simulation montrent que l'algorithme FW-WSC est plus performant que les deux autres approches les plus connus dans la littérature(SC-ESWOA et QWSC-GWO)en termes de temps de composition, et de valeur d'utilité.

Sur la base du travail réalisé, nous pouvons établir quelques perspectives de recherche. Nous envisageons de :

- Prendre en compte d'autres modèles de composition plus complexe (parallèle, conditionnelle, etc.).
- De faire une présélection affin de réduire l'espace de recherche.
- Calculer la complexité.
- Prendre en compte les corrélations de qualité de service.
- Implémentation de notre approche dans un environnement réel, en se basant sur une plate-forme d'expérimentation.

Bibliographie

[Aiello et al., 2005] Aiello, M., Frankova, G., and Malfatti, D. (2005). What's in an agreement ? an analysis and an extension of ws-agreement. In *International Conference on Service-Oriented Computing*, pages 424–436. Springer.

[Al-Masri and Mahmoud, 2008] Al-Masri, E. and Mahmoud, Q. H. (2008). Investigating web services on the world wide web. In *Proceedings of the 17th international conference on World Wide Web*, pages 795–804.

[Alrifai et al., 2010] Alrifai, M., Skoutas, D., and Risse, T. (2010). Selecting skyline services for qos-based web service composition. In *Proceedings of the 19th international conference on World wide web*, pages 11–20.

[Ardagna and Pernici, 2007] Ardagna, D. and Pernici, B. (2007). Adaptive service composition in flexible processes. *IEEE Transactions on software engineering*, 33(6) :369–384.

[ATMANI and CHERIFI, 2018] ATMANI and CHERIFI (2018). Evolutionary-based algorithms for qos-aware services composition in large-scale internet of things. Master's thesis in Computer Science, University of Bejaia.

[ATOUMI and BENSADI, 2018] ATOUMI and BENSADI (2018). Approche évolutionnaire pour la composition de services sensible à la qos dans l'internet des objets à large échelle. Master's thesis in Computer Science, University of Bejaia.

[BEKKOUCHE, 2018] BEKKOUCHE, A. (2018). *Vers une composition automatique des services web sémantiques*. phd' thesis in Computer Science, Université de Tlemcen-Abou Bekr Belkaid.

[BENKHALED and REGRAGUI, 2017] BENKHALED and REGRAGUI (2017). Selection de web services a base de la qualite de service. Master's thesis in Computer Science, University Tahar Moulay of SAIDA.

[Bhaskar et al., 2020] Bhaskar, B., Jatoth, C., Gangadharan, G., and Fiore, U. (2020). A mapreduce-based modified grey wolf optimizer for qos-aware big service composition. *Concurrency and Computation : Practice and Experience*, 32(8) :e5351.

[Bieberstein et al., 2006] Bieberstein, N., Bose, S., Fiammante, M., Jones, K., and Shah, R. (2006). *Service-oriented architecture compass : business value, planning, and enterprise roadmap.* FT Press.

[Birogul, 2019] Birogul, S. (2019). Hybrid harris hawk optimization based on differential evolution (hhode) algorithm for optimal power flow problem. *IEEE Access*, 7 :184468–184488.

[Canfora et al., 2005] Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L. (2005). An approach for qos-aware service composition based on genetic algorithms. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 1069–1075.

[Chen et al., 2015] Chen, Y., Huang, J., Lin, C., and Hu, J. (2015). A partial selection methodology for efficient qos-aware service composition. *IEEE Transactions on Services Computing*, 8(3) :384–397.

[Chollet, 2009] Chollet, S. (2009). *Orchestration de services hétérogènes et sécurisés.* PhD thesis.

[Curbera et al., 2002] Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. (2002). Unraveling the web services web : an introduction to soap, wsdl, and uddi. *IEEE Internet computing*, 6(2) :86–93.

[Deng et al., 2014] Deng, S., Wu, H., Hu, D., and Zhao, J. L. (2014). Service selection for composition with qos correlations. *IEEE Transactions on Services Computing*, 9(2) :291–303.

[Dodani, 2004] Dodani, M. H. (2004). From objects to services : A journey in search of component reuse nirvana. *J. Object Technol.*, 3(8) :49–54.

[Garg et al., 2013] Garg, S. K., Versteeg, S., and Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29(4) :1012–1023.

[Gavvala et al., 2019] Gavvala, S. K., Jatoth, C., Gangadharan, G., and Buyya, R. (2019). Qos-aware cloud service composition using eagle strategy. *Future Generation Computer Systems*, 90 :273–290.

[Hadjila, 2014] Hadjila, F. (2014). *Composition et interopération des services web sémantiques.* PhD thesis.

[Huo et al., 2015] Huo, Y., Zhuang, Y., Gu, J., Ni, S., and Xue, Y. (2015). Discrete gbest-guided artificial bee colony algorithm for cloud service composition. *Applied Intelligence*, 42(4) :661–678.

[Jatoth et al., 2018] Jatoth, C., Gangadharan, G., Fiore, U., and Buyya, R. (2018). Qos-aware big service composition using mapreduce based evolutionary algorithm with guided mutation. *Future Generation Computer Systems*, 86 :1008–1018.

[KABACHE and ZAIDI, 2018] KABACHE and ZAIDI (2018). Sélection et composition de services web avec respect des contraintes d'utilisateur et qualité de service. Master's thesis in Computer Science, University of Bejaia.

[Karimi and Babamir, 2017] Karimi, M. and Babamir, S. M. (2017). Qos-aware web service composition using gray wolf optimizer. *International Journal of Information and Communication Technology Research*, 9(1) :9–16.

[Kellert and Toumani, 2003] Kellert, P. and Toumani, F. (2003). Les web services sémantiques. *Web sémantique, Action spécifique*, 32.

[Khanouche et al., 2016] Khanouche, M. E., Amirat, Y., Chibani, A., Kerkar, M., and Yachir, A. (2016). Energy-centered and qos-aware services selection for internet of things. *IEEE Transactions on Automation Science and Engineering*, 13(3) :1256–1269.

[Khanouche et al., 2020] Khanouche, M. E., Gadouche, H., Farah, Z., and Tari, A. (2020). Flexible qos-aware services composition for service computing environments. *Computer Networks*, 166 :106982.

[Kumar and Purohit, 2016] Kumar, S. and Purohit, L. (2016). Exploring k-means clustering and skyline for web service selection. In *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, pages 603–607. IEEE.

[Li et al., 2020] Li, C., Li, J., and Chen, H. (2020). A meta-heuristic-based approach for qos-aware service composition. *IEEE Access*, 8 :69579–69592.

[Moghaddam and Davis, 2014] Moghaddam, M. and Davis, J. G. (2014). Service selection in web service composition : A comparative review of existing approaches. In *Web Services Foundations*, pages 321–346. Springer.

[Papazoglou, 2003] Papazoglou, M. P. (2003). Service-oriented computing : Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, pages 3–12. IEEE.

[Purohit and Kumar, 2019] Purohit, L. and Kumar, S. (2019). Clustering based approach for web service selection using skyline computations. In *2019 IEEE International Conference on Web Services (ICWS)*, pages 260–264. IEEE.

[Sheng et al., 2014] Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. (2014). Web services composition : A decade's overview. *Information Sciences*, 280 :218–238.

[Sun and Zhao, 2012] Sun, S. X. and Zhao, J. (2012). A decomposition-based approach for service composition with global qos guarantees. *Information Sciences*, 199 :138–153.

[Tan and Zhu, 2010] Tan, Y. and Zhu, Y. (2010). Fireworks algorithm for optimization. In *International conference in swarm intelligence*, pages 355–364. Springer.

[Tang and Xu, 2005] Tang, X. and Xu, J. (2005). Qos-aware replica placement for content distribution. *IEEE Transactions on parallel and distributed systems*, 16(10) :921–932.

[Tari et al., 2009] Tari, K., Amirat, Y., Chibani, A., and Yachir, A. (2009). Rule-based approach for automatic service composition in ubiquitous environment. In *The 6th International Conference On Ubiquitous Robots And Ambient Intelligence (URAI)*.

[Wagner et al., 2011] Wagner, F., Ishikawa, F., and Honiden, S. (2011). Qos-aware automatic service composition by applying functional clustering. In *2011 IEEE International Conference on Web Services*, pages 89–96. IEEE.

[Wang et al., 2017] Wang, S., Zhou, A., Yang, M., Sun, L., Hsu, C., and Yang, F. (2017). Service composition in cyber-physical-social systems. *IEEE Transactions on Emerging Topics in Computing*, 8 :82–91.

[Yachir, 2014] Yachir, A. (2014). *Composition dynamique de services sensibles au contexte dans les systèmes intelligents ambiants*. PhD thesis.

[Yachir et al., 2008] Yachir, A., Tari, K., Chibani, A., and Amirat, Y. (2008). Towards an automatic approach for ubiquitous robotic services composition. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3717–3724. IEEE.

[Yu and Bouguettaya, 2013] Yu, Q. and Bouguettaya, A. (2013). Efficient service skyline computation for composite service selection. *IEEE Transactions on Knowledge and Data Engineering*, 25(4) :776–789.

[Yue et al., 2020] Yue, Z., Zhang, S., and Xiao, W. (2020). A novel hybrid algorithm based on grey wolf optimizer and fireworks algorithm. *Sensors*, 20(7) :2147.

[Zeng et al., 2004] Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J., and Chang, H. (2004). Qos-aware middleware for web services composition. *IEEE Transactions on software engineering*, 30(5) :311–327.

[Zhao et al., 2017] Zhao, D., Zhou, Z., Ning, K., Duan, Y., and Zhang, L.-J. (2017). An energy-aware service composition mechanism in service-oriented wireless sensor networks. In *2017 IEEE International Congress on Internet of Things (ICIOT)*, pages 89–96. IEEE.

Résumé

De nos jours, l'utilisation des services web est indispensable par les entreprises pour rendre accessible leurs métiers et leurs données via le Web. Cependant il devient difficile de satisfaire l'exigence complexe d'un utilisateur par un seul service individuel ou atomique, ce qui conduit les concepteurs de coopérer entre les services simple existant pour créer un service composite.

Dans ce travail, nous avons proposé une approche de composition de services sensible à la Qos, prenant en compte les préférences des utilisateurs nommée (FW-SWC). La solution proposée est basée sur l'algorithme des feux d'artifices(FWA). La force de l'approche proposée réside dans sa capacité d'exploration qui lui permet de déterminer l'optimum le plus proche. De plus, dans cette approche la solution optimale est trouvée en un nombre réduit d'itération. Cela permet à l'algorithme de converger rapidement tout en retournant le meilleur résultat, ce qui entraîne aussi un temps d'exécution réduit.

Mots clés : Composition de Services ; Qualité de service(QoS) ; Exploration ; Exploitation ; Algorithme des feux d'artifice(FWA).

Abstract

Nowadays, the use of web services is essential by companies to make their businesses and their data accessible via the Web. However it becomes difficult to satisfy the complex requirement of a user by a single individual or atomic service, which leads designers to cooperate between existing single services to create a composite service.

In this work, we proposed QoS-aware web services composition approach, taking into account the preferences of users named(FW-SWC). The proposed solution is based on the Fireworks Algorithm (FWA). The strength of the proposed approach lies in its exploration capacity which allows it to determine the closest optimum. Moreover, in this approach the optimal solution is found in a reduced number of iterations. This allows the algorithm to converge quickly while returning the best result, which also results in reduced execution time.

Key words : Services composition ; Quality of Service(QoS) ; Exploration ; Exploitation ; Fireworks algorithm.