



Université de Bejaia  
Faculté des Sciences Exactes  
Département d'Informatique

# Mémoire de Master Recherche En Informatique

Option  
Système d'information Avancés

---

Proposition d'une approche de composition de services IoT sensible aux QoS

---

Réaliser par : *Mlle. ALLIK DALILA*

*Mlle. BELKACEMI KAMILIA*

**Soutenu le 18 Septembre 2024**

**Évaluée par :**

Mme. CHIBANI SADOUKI SAMIA	Présidente
Mme.GASMI BADRINA	Examineur
Mme.AIT HACENE SOUHILA	Encadrante

**Promotion 2023/2024**

## Remerciements

Tout d'abord, nous souhaitons exprimer notre infinie gratitude envers Dieu, qui nous a accordé la force, la patience et la persévérance nécessaires pour mener à bien ce projet.

Nous tenons à souligner notre reconnaissance mutuelle pour le travail en binôme que nous avons réalisé. Nos efforts conjoints, notre détermination, et notre soutien réciproque ont été essentiels à l'aboutissement de ce travail. Ensemble, nous avons su surmonter les défis et les difficultés rencontrés au cours de cette aventure académique.

Nos plus sincères remerciements vont à notre encadrante, Mme AIT HACENE Souhila. Son dévouement, ses conseils avisés, et sa disponibilité constante ont été des éléments clés dans la réussite de notre projet. Son accompagnement tout au long de ce parcours a été inestimable, et nous lui en sommes profondément reconnaissants.

Nous tenons également à exprimer notre gratitude aux membres du jury, qui ont accepté d'évaluer notre travail. Leur expertise et leurs commentaires constructifs seront pour nous une source précieuse d'apprentissage et de perfectionnement.

Enfin, nous souhaitons remercier tous les enseignants de l'Université de Béjaïa qui ont contribué à notre formation au fil des années. Leurs enseignements et leur encadrement ont façonné nos compétences et notre savoir, nous permettant d'arriver là où nous sommes aujourd'hui. Leur dévouement à notre éducation mérite toute notre reconnaissance.

# Dédicace

## **À ma chère maman,**

Maman, tu as toujours été mon roc, mon pilier, ma lumière dans les moments les plus sombres. Tes sacrifices, ton amour inconditionnel et ta présence constante à mes côtés m'ont donné la force de surmonter chaque obstacle. Cette réussite est autant la tienne que la mienne, car sans toi, je n'y serais jamais arrivée. Je te dédie ce travail avec toute la reconnaissance et l'amour que mon cœur peut contenir.

## **À mon père,**

Papa, ta sagesse, ta patience et ton soutien indéfectible m'ont permis de rester sur la bonne voie. Tes conseils avisés et ton calme rassurant m'ont toujours donné la confiance nécessaire pour avancer. Ce travail, je te le dédie avec toute mon affection.

## **À ma sœur Sara, et à mes frères Sidahmed et Yanis Ghiles,**

Vous êtes ma force, mes complices de toujours. Votre amour et vos encouragements ont été une source précieuse de motivation. Merci pour tout ce que vous avez fait pour moi. Je vous dédie ce travail avec tout mon amour.

## **À mon grand-père, EL HADJ HAMIMI,**

Bien que tu ne sois plus parmi nous, je sais que tu aurais été fier de ta petite-fille aujourd'hui. Que Dieu t'accueille dans son vaste paradis, et que ta mémoire continue d'être une source d'inspiration pour moi. Je te dédie ce travail en hommage à ta mémoire.

## **À mon oncle maternel, Karim Haddad,**

Mon cher oncle, tu as toujours été là pour me soutenir, me conseiller, et m'encourager dans tous les moments importants de ma vie. Ta présence constante et ton aide précieuse ont grandement contribué à mon succès. Je te dédie ce travail avec toute ma reconnaissance.

## **À mon cher Wajdy,**

Wajdy, tu es une source inestimable de soutien et de réconfort dans ma vie. Ta présence constante, tes encouragements et ton amour m'ont aidée à traverser les moments difficiles et à célébrer les réussites. Je te dédie ce travail avec toute ma gratitude et mon affection.

## **À toute ma famille BELKACEMI,**

Je vous dédie ce travail, car vous avez tous contribué à ma réussite d'une manière ou d'une autre. Vos encouragements, votre amour, et votre soutien m'ont porté jusqu'ici.

**À mes amies de l'université :Ouardia, Wiam,Khalida ET Naima et Dina ,**

Vous êtes devenues bien plus que des amies. Vous êtes mes sœurs de cœur, mes confidents, et mes partenaires de galère. Je vous dédie ce travail, en espérant que notre amitié continue de grandir avec le temps.

**À ma binôme, Dalila,**

Dalila, merci pour ta patience, ton soutien, et ta collaboration tout au long de ce projet. Nous avons formé une équipe formidable et ce travail est le fruit de nos efforts conjoints.

**À toutes les personnes qui m'ont aidée et soutenue,** Je n'oublie personne dans ce moment de gratitude. Que ce soit par un mot d'encouragement, un geste de soutien ou simplement votre présence, vous avez tous contribué à ce moment. Je vous dédie ce travail avec reconnaissance.

**Et enfin, à moi-même,**

À moi, pour mes efforts, pour ma persévérance, et pour avoir tenu bon malgré tout ce que j'ai traversé. Ce travail est le résultat de ma détermination et de ma volonté de réussir, malgré les épreuves. Que cette réalisation soit un rappel de ma force intérieure et de ce que je suis capable d'accomplir.

Mlle.BELKACEMI KAMILIA

# Dédicace

C'est avec une profonde gratitude que je dédie ce travail à :

**Mon cher père, Allik Saïd,**  
qui m'a toujours soutenu sans compter et encouragé à devenir meilleur,  
tout en me donnant sans limites.

**Ma chère mère, Dahmani Henia,**  
dont la lumière et les prières m'ont guidé à chaque étape  
de mon cheminement.

**Mes sœurs, Kenza et Taous,**  
ainsi que **mes grandes sœurs, Lila et Noria,** pour leur affection et soutien constants.  
**Mes cousins Tarik et Khaled,** qui sont toujours là pour moi lorsque j'ai besoin d'aide.

**Ma famille entière,** pour leur appui inconditionnel.

Je tiens également à remercier **ma meilleure amie Lydia,** qui a toujours été à mes côtés,  
**mon amie d'enfance Ferroudja** qui m'encourage toujours, et **ma copine Rania,** avec  
qui j'ai passé mes moments à l'université et qui me motive toujours.

**Ma binôme, Belkacemi Kamilia,**  
à qui je souhaite plein de succès et de bonheur.

**Moi-même,** pour ne jamais avoir abandonné, et pour la persévérance et l'effort  
déployés tout au long de cette aventure.

À tous ceux qui ont contribué à ma réussite, un immense merci.

Mlle.ALLIK DALILA

---

## Résumé

La composition de services IoT est devenue un enjeu crucial dans le monde actuel, où la performance et la gestion des coûts sont des facteurs déterminants. Ce projet propose une approche efficace de composition des services IoT, tenant compte de divers critères de qualité de service (QoS) tels que le temps d'exécution, le coût, la crédibilité et la fiabilité. L'objectif principal est de simplifier le processus de sélection des meilleures combinaisons de services IoT tout en optimisant les valeurs de QoS. Pour y parvenir, des méthodes d'optimisation, telles que l'algorithme Grey Wolf Optimization (GWO), combiné à K-Means pour le regroupement des données, sont utilisées.

Ces techniques sont appliquées dans le cadre d'une étude de cas concrète : une sonnette vidéo connectée. L'objectif est d'améliorer la performance du service IoT tout en réduisant les coûts d'exploitation.

**Mots clés** : IoT, composition de services, qualité de service, Grey Wolf Optimization, K-Means, clustering.

## Abstract

The composition of IoT services has become a crucial issue in today's world, where performance and cost management are determining factors. This project proposes an efficient approach to IoT service composition, taking into account various quality of service (QoS) criteria such as execution time, cost, credibility and reliability. The main objective is to simplify the process of selecting the best combinations of IoT services while optimizing QoS values. To achieve this, optimization methods such as the Grey Wolf Optimization (GWO) algorithm, combined with KMeans for data clustering, are used. These techniques are applied to a concrete case study : a connected video doorbell. The aim is to improve IoT service performance while reducing operating costs.

**Keywords** : IoT, service composition, optimization, quality of service, Grey Wolf Optimization, KMeans, clustering.

# Table des matières

Liste des figures	V
Liste des tables	VI
Liste des algorithmes	VII
Liste des Abréviations	IX
<b>1 Chapitre 1 : Généralités sur l'Internet des Objets (IoT)</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Définition de l'IoT . . . . .	3
1.3 Historique de l'IoT . . . . .	3
1.4 Architecture de l'IoT . . . . .	4
1.4.1 La couche de perception . . . . .	4
1.4.2 La couche de réseau . . . . .	4
1.4.3 La couche d'application . . . . .	5
1.5 Les composants d'un système IoT . . . . .	5
1.6 Domaines d'application de l'IoT . . . . .	7
1.6.1 Villes intelligentes . . . . .	7
1.6.2 Maison intelligente . . . . .	7
1.6.3 Transport intelligent . . . . .	7
1.6.4 Industrie et fabrication intelligente . . . . .	7
1.6.5 Santé connectée . . . . .	7
1.7 Défis de l' IoT . . . . .	8
1.8 Avantages de l'IoT . . . . .	9
1.9 Développements futurs de l'IoT . . . . .	9
1.10 Conclusion . . . . .	11
<b>2 Chapitre 2 La sélection de services IoT sensible aux QoS pendant le processus de composition</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 Définitions d'un service IoT . . . . .	12
2.3 Définition de la composition des services IoT . . . . .	12
2.4 Processus de la composition de services IoT tenant compte du QoS . . . . .	13

2.5	Concepts de base liés à la composition des services IoT sensibles aux QoS . . . . .	14
2.5.1	Service candidat (CS) . . . . .	14
2.5.2	Service abstrait (AS) . . . . .	14
2.5.3	Services composites abstraits (ACS) . . . . .	15
2.5.4	Service composite concret (CCS) . . . . .	15
2.5.5	Critères non fonctionnels (QoS) . . . . .	15
2.5.6	Vecteur de QoS d'un service concret $QoS(CS_i)$ . . . . .	15
2.5.7	Vecteur d'attributs QoS d'une composition $V_{QoS}(comp)$ . . . . .	16
2.6	Approches de sélection des services sensible aux Qos . . . . .	16
2.6.1	Sélection locale . . . . .	17
2.6.2	Sélection globale . . . . .	17
2.7	Approches et méthodes existantes . . . . .	17
2.7.1	Approches basées sur les méta-heuristique . . . . .	17
2.7.2	Apprentissage automatique . . . . .	18
2.8	Conclusion . . . . .	20
<b>3</b>	<b>Chapitre 3 État de l'art</b>	<b>21</b>
3.1	Introduction . . . . .	21
3.2	Analyse Documentaire . . . . .	21
3.3	Étude comparative entre les approches . . . . .	26
3.4	Discussion . . . . .	32
3.5	Conclusion . . . . .	33
<b>4</b>	<b>Chapitre 4 : Approche proposée : K-means with Grey Wolf Optimization Algorithm</b>	<b>34</b>
4.1	Introduction . . . . .	34
4.2	Motivation . . . . .	34
4.3	L'algorithme K-Means . . . . .	34
4.4	Grey Wolf Optimization (Optimisation par Loup Gris) . . . . .	35
4.4.1	La recherche de proies . . . . .	36
4.4.2	L'encerclement de la proie . . . . .	36
4.4.3	La chasse de la proie . . . . .	37
4.4.4	L'attaque de proie . . . . .	38
4.5	Fonctionnement de l'approche proposée KM-GWO . . . . .	39

4.5.1	Classification des services candidats via K-means . . . . .	40
4.5.2	Application de GWO . . . . .	41
4.6	Validation de l'approche . . . . .	47
4.6.1	Environnement et outils de développement . . . . .	47
4.6.2	Description de Dataset . . . . .	49
4.6.3	Prétraitement des données du <i>IoTS<sub>D</sub>dataset</i> . . . . .	50
4.6.4	Scénario de validation de KM-GWO . . . . .	53
4.7	Application de KM-GWO . . . . .	54
4.8	Application de GWO pour rechercher une composition proche de l'optimale	57
4.8.1	Initialisation de la Population . . . . .	57
4.8.2	Évaluation des composition (Fitness) . . . . .	60
4.9	Recherche d'une composition quasi-optimal via GWO . . . . .	61
4.10	Conclusion . . . . .	63
<b>5</b>	<b>Conclusion Générale</b>	<b>64</b>

## Table des figures

1	Architecture de l'IoT . . . . .	5
2	Composants d'un système IoT . . . . .	6
3	IoT analytics . . . . .	10
4	Composition des services IoT . . . . .	13
5	Processus de la composition des services IoT en fonction de la QoS . . . . .	14
6	Categories-d'apprentissage-automatique . . . . .	20
7	K-means . . . . .	35
8	Hierarchy of grey wolf . . . . .	36
9	Mise à jour de position par les loups gris . . . . .	38
10	L'algorithme GWO original . . . . .	39
11	Digramme de flux de l'algorithme GWO . . . . .	46
12	IoT Dataset . . . . .	50
13	Import d'un Dataset . . . . .	51
14	Dataset avant la normalisation . . . . .	52
15	Dataset après la normalisation . . . . .	52
16	plan de composition pour le scénario de KM-GWO . . . . .	53
17	Classification initiale avec K-means . . . . .	55
18	Répartition des services IoT en fonction de leur QoS-somme . . . . .	56
19	Répartition des Services dans les Clusters . . . . .	56
20	Score de Silhouette . . . . .	57
21	Initialisation de la population . . . . .	58
22	Initialisation de la population2 . . . . .	58
23	Initialisation de la population3 . . . . .	59
24	Résultat d'agrèger les valeurs de QoS . . . . .	59
25	Normalisation après l'agrégation . . . . .	60
26	Evaluation des compositions avec fitness . . . . .	61
27	Graphe de fitness en fonction de services candidats . . . . .	61
28	Meilleure solution . . . . .	62
29	Évaluation des valeurs de fitness des KM-GWO et de GWO traditionnel. . . . .	63

## Liste des tableaux

1	Fonctions d'agrégation des QoS pour le calcul du $V_{QoS}(comp)$ . . . . .	16
2	Comparaison des méthodes de sélection de services IoT . . . . .	30
3	Critères traités dans chaque article . . . . .	32
4	Services candidats disponibles pour chaque service abstrait dans le scénario de composition . . . . .	54
5	Évaluation de Valeurs de fitness de KM-GWO et de GWO . . . . .	62

## List of Algorithms

1	Algorithme K-means	.....	41
2	Algorithme de GWO	.....	45

## Liste des Abréviations

<b>IoT</b>	Internet des Objets
<b>QoS</b>	Qualité de Service
<b>VQoS</b>	Vecteur de Qualité de Service
<b>CS</b>	Service Candidat
<b>AS</b>	Service Abstrait
<b>ACS</b>	Services Composites Abstraits
<b>CCS</b>	Service Composite Concret
<b>MA</b>	Multi-Agents
<b>IaaS</b>	Infrastructure as a Service
<b>VMs</b>	Virtual Machines
<b>WS</b>	Web Services
<b>AI</b>	Artificial Intelligence
<b>PSO</b>	Particle Swarm Optimization
<b>WSC</b>	Web Services Composition
<b>WWM</b>	World Wide Minds
<b>GA</b>	Genetic Algorithm
<b>GWO</b>	Grey Wolf Optimizer
<b>KM-GWO</b>	K-means and Grey Wolf Optimizer
<b>NSGA</b>	Non-dominated Sorting Genetic Algorithm
<b>MCGDM</b>	Multi-Criteria Group Decision-Making
<b>TOPSIS</b>	Technique for Order Preference by Similarity to Ideal Solution
<b>DM</b>	Decision Maker
<b>OWA</b>	Ordered Weighted Averaging Aggregation
<b>DEOSA</b>	Dynamic Effect-Oriented Service Selection Algorithm
<b>MOEA</b>	Multi-Objective Evolutionary Algorithm
<b>ISFLA</b>	Improved Shuffled Frog-Leaping Algorithm
<b>EQSA</b>	Energy-centered and QoS-aware Services Selection Algorithm
<b>MRED</b>	Multi-Agent Reactive Estimation Distribution
<b>RBM</b>	Restricted Boltzmann Machines
<b>EDA</b>	Estimation of Distribution Algorithms
<b>A</b>	Vecteur de Coefficients
<b>a</b>	Vecteur qui Diminue Linéairement de 2 à 0 au Cours des Itérations
<b>C</b>	Vecteur de Coefficients
<b>D</b>	Distance entre un Loup Gris et la Proie
<b><math>r_1</math></b>	Vecteur Aléatoire Compris entre 0 et 1

## Liste des Abréviations

---

$r_2$	Vecteur Aléatoire Compris entre 0 et 1
$X_p$	Position de la Proie
$X_i$	Position du Loup Gris à l'Itération $t + 1$

## Introduction Générale

L'Internet des objets (IoT) constitue une avancée technologique majeure qui relie des objets physiques à l'univers numérique. Cette connexion permet aux objets de collecter, traiter et partager des données, ouvrant la voie à de nouvelles possibilités dans divers domaines, tels que les maisons intelligentes, les villes connectées, la santé, la sécurité, la logistique et l'industrie. Au cœur de cette révolution se trouve le service IoT, qui se définit comme une fonctionnalité spécifique offerte par un objet connecté, capable de recueillir, traiter et transmettre des informations pour répondre à des besoins particuliers. La composition de ces services devient alors essentielle pour créer des solutions répondant à des requêtes complexes.

La composition de services IoT est primordiale pour répondre aux besoins variés des utilisateurs. En effet, un seul service IoT ne suffit généralement pas à satisfaire pleinement ces besoins. Ainsi, l'approche de composition permet de combiner plusieurs services pour concevoir des applications personnalisées. La croissance du nombre d'objets intelligents dans l'environnement IoT a conduit à la création d'un grand nombre de services dotés de fonctionnalités similaires, mais avec des propriétés de la qualité de services variables. Cela rend complexe la satisfaction des besoins des utilisateurs et pose un problème NP-difficile pour la sélection des services à inclure dans une composition. En effet, l'objectif est de sélectionner les services les plus adaptés qui répondent à la fois aux exigences fonctionnelles et non fonctionnelles à savoir les préférences des utilisateurs en matière de QoS.

Notre contribution consiste à proposer une approche de composition de services IoT sensible à la qualité de services, cette approche qui combine k-means avec l'algorithme de grey wolf optimizer GWO est nommé KM-GWO . KM-GWO a pour objectif de rechercher une solution quasi optimale en choisissant la meilleure composition de services IoT afin de répondre aux attentes des utilisateurs dans un espace de recherche réduit.

Le mémoire est organisé en trois chapitres principaux. Le premier chapitre introduit la technologie de l'IoT, définissant ses concepts fondamentaux et composants principaux. Il présente l'architecture de l'IoT, ses domaines d'application, ainsi que les avantages et les défis associés à cette technologie. Le deuxième chapitre se concentre sur les définitions essentielles liées à la compréhension du problème de composition de services. Il détaille les étapes du processus de composition des services IoT et explore les concepts clés du Machine Learning et des méta-heuristiques. Le troisième et dernier chapitre présente l'approche proposée, nommée K-means and Grey Wolf Optimizer Algorithm (KM-GWO). Ce

chapitre présente l'approche ainsi que son évaluation en comparaison avec l'algorithme GWO original, fournissant des résultats qui démontrent l'efficacité de KM-GWO. Nous clôturons le mémoire par une conclusion générale et quelques perspectives sur les travaux futurs.

### Chapitre 1 : Généralités sur l'Internet des Objets (IoT)

#### 1.1 Introduction

Ce chapitre est dédié à l'exploration des généralités sur l'Internet des objets, une technologie révolutionnaire qui connecte un vaste réseau d'objets physiques à travers le monde. Nous allons d'abord présenter une définition de l'IoT et analyser ses composants principaux. Ensuite, nous exposerons son architecture et ses divers domaines d'application. Enfin, nous citerons les évolutions à venir de cette technologie en constante évolution et les défis de l'IoT.

#### 1.2 Définition de l'IoT

L'Internet des objets est un réseau mondial d'objets physiques connectés à Internet. Dans l'IoT les dispositifs connectés sont appelés objets intelligents et ils sont équipés de capteurs et d'actionneurs ainsi que de capacité de collecte et d'échange de données [42]. Les objets IoT comprennent les appareils de la vie quotidienne (domotique, montres de fitness, etc.), les dispositifs médicaux, les machines agricoles, les chaînes logistiques. L'IoT permet de connecter tout élément capable de transférer des données sur un réseau, sans intervention nécessaire d'humains. Cependant, une interaction homme-machine est possible pour configurer, paramétrer ou accéder aux informations[45].

L'IoT est une technologie en pleine expansion qui rend notre monde plus connecté, intelligent et efficace. En s'intégrant de manière transparente à notre environnement, l'IoT promet de simplifier nos tâches quotidiennes, contribuant à rendre notre vie plus sûre et plus satisfaisante [39].

#### 1.3 Historique de l'IoT

L'Internet des objets, né de concepts futuristes des années 1990 et trouvant ses racines dans les années 1970, a connu une évolution révolutionnaire. L'idée d'appareils connectés remonte à cette époque, notamment avec l'exemple emblématique du premier objet connecté : un distributeur de Coca-Cola modifiée à l'Université de Carnegie Mellon au début des années 1980. Ce distributeur permettait aux étudiants de surveiller à distance, via un réseau précurseur de l'internet, l'état des stocks et la température des boissons, illustrant ainsi les premières étapes concrètes de l'Internet des Objets [61].

Le terme "Internet des objets" a été inventé par Kevin Ashton lors d'une présentation

chez Proctor & Gamble en 1999. Il est l'un des fondateurs du laboratoire de reconnaissance automatique du Massachusetts Institute of Technology. Il a été l'un des pionniers de la technologie RFID (utilisée dans les détecteurs de codes-barres) dans le domaine de la gestion de la chaîne d'approvisionnement. Il a également fondé ZENSI, une société qui fabrique des technologies de détection et de surveillance de l'énergie [40].

Le concept d'interconnexion des machines a été mis en avant par Neil Gershenfeld, professeur du Massachusetts Institute of Technology (MIT), dans son ouvrage "When Things Start to Think" paru en 1999. L'IoT s'est rapidement développé autour de la communication machine to machine (M2M), qui décrit la capacité des appareils à se connecter et à communiquer les uns avec les autres sans interaction humaine [5].

L'Internet des objets d'aujourd'hui est un vaste réseau de capteurs équipés de milliards d'objets intelligents, unissant les individus, les systèmes informatiques et les applications dans un but commun : partager et collecter des données essentielles à notre quotidien et à nos avancées technologiques [5].

### 1.4 Architecture de l'IoT

L'architecture de l'IoT n'est pas standardisée en raison de la grande variété de technologies utilisées dans ces environnements. Cependant, il existe une configuration typique qui intègre les divers composants d'un système IoT afin d'assurer la collecte, le stockage et le traitement des données émanant des objets via les capteurs, tout en permettant aux actionneurs de ces objets d'exécuter les commandes transmises par le biais d'une application utilisateur. Cette configuration est souvent structurée en trois couches distinctes à savoir : la couche perception, la couche réseau, et la couche application [53].

Dans la suite de cette section, nous allons expliquer le fonctionnement et le rôle de ces couches.

#### 1.4.1 La couche de perception

La couche de perception joue un rôle crucial dans la transformation des signaux analogiques en données numériques et vice versa. Elle englobe une variété d'éléments qui agissent comme des intermédiaires entre le monde physique et numérique, comme les capteurs, les actionneurs et les machines connectées.

#### 1.4.2 La couche de réseau

La couche réseau relie les appareils à d'autres objets intelligents, serveurs et dispositifs réseau. Elle assure également le transfert et le traitement de toutes les données ainsi que

la collecte des données générées par des capteurs.

### 1.4.3 La couche d'application

La couche application, également appelée couche de services, propose aux utilisateurs finaux une variété de services et de fonctionnalités, notamment la visualisation des données et la prise de décisions basées sur les informations collectées par les objets connectés [20].

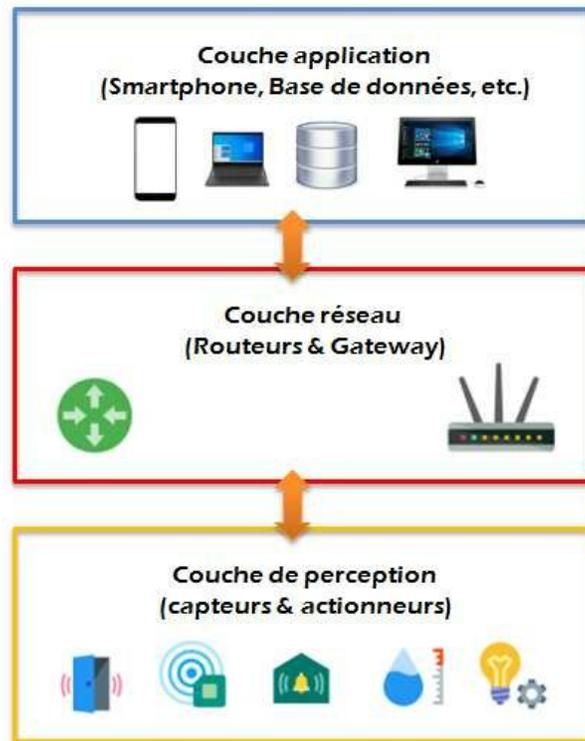


FIGURE 1 – Architecture de l'IoT [9]

## 1.5 Les composants d'un système IoT

L'IoT s'appuie sur divers composants pour créer des systèmes intelligents capables de capturer et de traiter des données du monde réel. Dans cette section, nous examinerons les principaux éléments qui composent un système IoT :

1. **Objets connectés** : Ce sont les éléments centraux de l'IoT, équipés de capteurs et de microprocesseurs pour collecter et transmettre des données environnementales telles que la température, la pression et le mouvement [18].

2. **Réseau Local** : Les réseaux locaux sans fil (Wireless Sensor Network - WSN), réseaux locaux (Local Area Network - LAN) et réseaux personnels (Personal Area Network - PAN) permettent la connexion des capteurs et actionneurs aux passerelles. Ce sont des technologies de communication locales.
3. **Passerelle (Gateway - FOG/EDGE)** : Ce niveau comprend des passerelles qui assurent le lien entre les réseaux locaux et Internet. Elles permettent de gérer les données à la périphérie (Edge) avant de les envoyer au cloud. Cette approche permet d'effectuer des traitements locaux (Fog Computing) pour une réponse rapide.
4. **Internet ou LPWAN** : Ici, la passerelle envoie les données via Internet ou des réseaux à faible consommation (Low-Power Wide-Area Networks - LPWAN) tels que LoRaWAN, SigFox, ou des technologies cellulaires comme le NB-IoT, LTE-M, 3G et 2G. Ces réseaux permettent de transporter les données collectées par les capteurs sur de longues distances, souvent vers un serveur cloud.
5. **Plateformes Cloud** : Les données transitent ensuite vers un serveur cloud, où elles sont traitées et stockées. Le cloud utilise des plateformes de traitement de données, des serveurs web et des gestionnaires de réseaux machine-to-machine (M2M). Ces plateformes permettent une gestion centralisée des informations IoT.
6. **Applications et services** : Ils exploitent les données collectées pour proposer des services innovants, répartis en plusieurs catégories. La première catégorie est constituée des services basés sur l'identité, qui incluent la majorité des appareils en temps réel. Ensuite, nous avons les services d'agrégation d'informations, qui rassemblent les données brutes des capteurs dans le monde réel et les relient à des applications IoT adaptées. Les services collaboratifs tirent parti des données recueillies pour réaliser des analyses et soutenir la prise de décision. Enfin, les services ubiquitaires visent à permettre aux systèmes collaboratifs de fonctionner partout et à tout moment, répondant ainsi aux besoins des clients [ia-IoT].

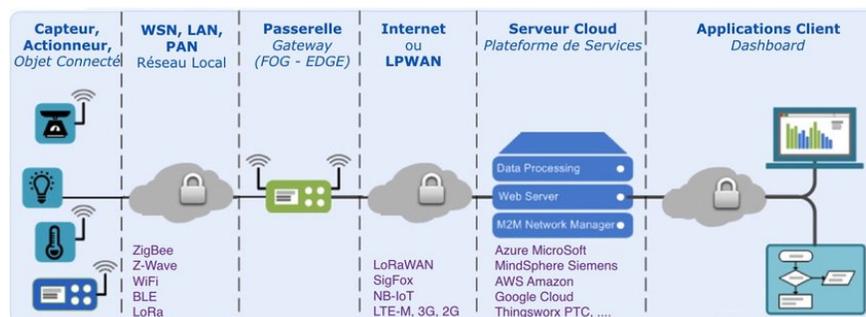


FIGURE 2 – Composants d'un système IoT [7]

### 1.6 Domaines d'application de l'IoT

L'IoT couvre divers aspects de notre vie quotidienne en proposant des solutions novatrices. Ces solutions transforment nos habitudes en améliorant notre qualité de vie. Dans cette section, nous exposons les principaux domaines d'application de l'IoT :

#### 1.6.1 Villes intelligentes

Les villes intelligentes utilisent l'IoT pour améliorer la qualité de vie de leurs habitants en utilisant des capteurs de stationnement qui guident vers un emplacement libre, des lampadaires connectés qui s'adaptent à la luminosité ambiante et des bacs à déchets qui envoient un signal lorsqu'ils sont pleins.

#### 1.6.2 Maison intelligente

Les maisons intelligentes utilisent la technologie pour automatiser et optimiser divers aspects de la maison afin d'améliorer le confort et la sécurité, notamment à travers des systèmes d'éclairage intelligents, de chauffage et de climatisation connectés, et des systèmes de sécurité intelligents.

#### 1.6.3 Transport intelligent

Le domaine du transport tire profit de nombreuses applications IoT tel que les parcs de voitures, de camions, de navires et de trains transportant des marchandises peuvent être réacheminés en temps réel en fonction des conditions météorologiques, de la disponibilité des véhicules ou des conducteurs, grâce aux données fournies par les capteurs IoT [45].

#### 1.6.4 Industrie et fabrication intelligente

Les industriels peuvent obtenir un avantage concurrentiel en surveillant leurs chaînes de production grâce aux capteurs. Ces capteurs détectent les pannes à venir et peuvent prédire quand la production sera affectée. Grâce aux alertes des capteurs, les industriels peuvent vérifier rapidement l'équipement ou l'enlever de la chaîne de production pour le réparer [45].

#### 1.6.5 Santé connectée

La santé connectée utilise des dispositifs IoT pour surveiller et améliorer la santé et le bien-être des individus. Cela inclut des montres intelligentes, des tensiomètres connectés,

des glucomètres connectés et des applications de télésanté, qui permettent aux patients de mieux contrôler leur santé, d'accéder à des diagnostics plus rapides et à des traitements personnalisés.

### 1.7 Défis de l' IoT

L'évolution rapide de l'IoT a ouvert de nouvelles perspectives passionnantes, mais elle s'accompagne également de défis significatifs. En explorant ces défis, nous pouvons mieux comprendre les obstacles à surmonter pour garantir le développement durable et sécurisé de l'IoT.

#### 1. Sécurité des appareils IoT

- Les appareils IoT sont souvent vulnérables aux attaques car les failles de sécurité ne sont pas corrigées, ou encore les appareils peuvent ne pas être mis à jour, ce qui en fait des cibles faciles pour les hackers [10].
- Transmission non sécurisée des données Les données échangées entre les appareils IoT et les serveurs peuvent ne pas être sécurisées, car ils utilisent des protocoles de communication qui ont été développés sans une sécurité robuste comme priorité principale [10].
- La diversité des appareils IoT connectés à différents réseaux soulève un défi, car il est compliqué d'appliquer des protocoles de sécurité uniformes à l'ensemble du système IoT en raison de sa diversité et de sa complexité.

#### 2. Complexité de l'écosystème IoT

- L'hétérogénéité et l'interconnexion des composants IoT compliquent la sécurisation globale, chaque composant pouvant présenter des protocoles, des interfaces et des niveaux de sécurité distincts [21].
- Le processus d'intégration complexe et coûteux des applications IoT dans les systèmes d'information existants nécessite des ajustements et des mises à niveau.
- Le manque de standardisation dans l'écosystème IoT est principalement dû à l'absence de règles ou de normes universellement acceptées pour la communication et l'interaction entre les différents appareils et systèmes IoT [21].
- Le défi de l'interopérabilité dans l'IoT concerne la capacité des différents appareils à fonctionner ensemble de manière transparente, impliquant l'établissement de normes et de protocoles partagés pour garantir une communication fluide entre les dispositifs de divers fabricants [16].

#### 3. Durabilité et gestion du cycle de vie

- La multiplication rapide des appareils IoT crée de plus en plus de déchets électroniques, ce qui pose des problèmes pour l'environnement et leur gestion.
  - La plupart des appareils IoT contiennent des matériaux qui peuvent être recyclés, mais le recyclage efficace de ces composants peut être difficile en raison de leur petite taille et de leur complexité [14].
  - Les appareils IoT peuvent parfois consommer beaucoup d'énergie, ce qui pose des défis en matière de durabilité et d'efficacité énergétique [14].
4. L'interface de l'utilisateur (UX/UI : User Experience/User Interface)
- Dans l'IoT concerne la création d'interfaces simples et pratiques pour les utilisateurs. Il faut concevoir des interfaces faciles à utiliser et à mettre à jour pour que les gens adoptent facilement les appareils IoT et les utilisent au quotidien [30].
  - Une intégration fluide des appareils : les différents appareils connectés doivent pouvoir interagir de manière fluide et transparente.

### 1.8 Avantages de l'IoT

Dans cette section nous allons présenter certains avantages de l'IoT à savoir :

1. **Amélioration de l'efficacité opérationnelle** : L'IoT simplifie les tâches répétitives, ce qui améliore la performance des activités commerciales [2] .
2. **Optimisation des ressources** : Grâce à la collecte en temps réel de données, l'IoT permet une approche plus adaptée [2].
3. **Meilleure prise de décision** : En fournissant des données précises et pertinentes, l'IoT assiste les entreprises dans la prise de décisions plus éclairées [2].
4. **Amélioration de la qualité de vie** : Dans des domaines tels que la santé et la domotique, l'internet des objets offre des solutions novatrices qui simplifient la vie.
5. **Innovation et nouveaux modèles commerciaux** : L'internet des objets génère de nouvelles idées et de nouveaux produits, créant ainsi de nouvelles possibilités et opportunités.
6. **Sécurité améliorée** : l'IoT renforce la protection des données avec des mesures de sécurité avancées, assurant ainsi une tranquillité d'esprit accrue [32].

### 1.9 Développements futurs de l'IoT

Le futur de l'IoT s'annonce plein de promesses, avec une croissance et des avancées technologiques remarquables. Dans cette section, nous discuterons de ces avancées.

## 1. Croissance explosive

- Dans un an, plus de 41 milliards d'appareils IoT devraient être déployés, entraînant une explosion des données et propulsant les opérations informatiques et l'analyse des données vers le bord du réseau. Cette expansion massive des appareils connectés contribuera à créer des environnements intelligents intégrant des technologies numériques pour améliorer notre quotidien [17].

## 2. Edge Computing

- Le passage à l'Edge Computing Permettra le traitement des données au niveau de l'appareil lui-même, améliorant la rapidité, la fiabilité, la confidentialité et la sécurité.
- Cette approche réduit les coûts de communication et de stockage inutiles tout en utilisant l'apprentissage automatique et l'IA pour détecter les tendances dans les données [36].

## 3. Prévisions du marché

- Le marché mondial de l'IoT devrait atteindre 3352,97 milliards de dollars d'ici 2030, avec un taux de croissance annuel composé d'un pourcentage de 26,1 [34].
- Ces avancées ouvrent de nouvelles opportunités commerciales, des applications innovantes et des sources de revenus supplémentaires grâce à une interaction intuitive entre l'homme et la machine [62].

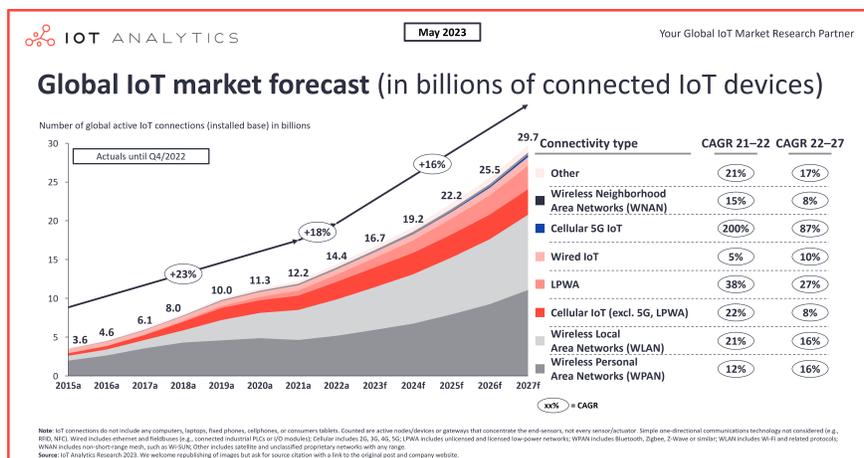


FIGURE 3 – IoT analytics [52]

### **1.10 Conclusion**

Ce chapitre a offert une analyse de la technologie IoT, en abordant ses différents composants, son architecture, ses domaines d'application, ainsi que certains avantages de l'IoT et son développement futur. Nous avons également examiné certains défis liés à son utilisation. Dans le prochain chapitre, nous explorerons le concept de composition des services IoT et plus particulièrement sur la sélection de services IoT. Nous examinerons également certains travaux de la littérature qui traite la sélection des services sensibles à la qualité de service.

## Chapitre 2 La sélection de services IoT sensible aux QoS pendant le processus de composition

### 2.1 Introduction

Dans ce chapitre, nous abordons les étapes clés du processus de sélection et de composition des services IoT, en mettant en lumière les défis liés à l'intégration harmonieuse des objets connectés. Nous introduisons les concepts fondamentaux tels que les services candidats, abstraits, et composites, ainsi que les vecteurs de QoS associés. Nous explorons ensuite les approches et techniques pour la sélection des services en tenant compte des critères de qualité de service, y compris les méthodes locales, globales, méta-heuristiques et d'apprentissage automatique.

Ces techniques sont essentielles pour surmonter les défis complexes liés à la sélection des services IoT et garantir la flexibilité, la qualité de service et l'efficacité opérationnelle des solutions IoT.

### 2.2 Définitions d'un service IoT

Un service IoT se définit comme une entité logicielle ou informatique, conçue pour fournir des fonctionnalités spécifiques en lien avec l'IoT et ses utilisateurs. Il se compose de quatre éléments essentiels : les entrées, les sorties, les fonctionnalités (propriétés fonctionnelles) et la qualité de service (QoS, propriétés non fonctionnelles). Les entrées désignent les données ou informations nécessaires à l'opération du service IoT, généralement collectées via des capteurs ou dispositifs connectés. Les sorties, quant à elles, sont les résultats produits en réponse à ces entrées, souvent sous forme d'informations ou d'actions communiquées à d'autres systèmes ou appareils. Les fonctionnalités définissent les capacités spécifiques du service, telles que la surveillance, le contrôle ou l'analyse en temps réel, qui exploitent les données des objets connectés. Enfin, la QoS couvre des aspects essentiels, comme la fiabilité, la disponibilité, les performances, et d'autres critères qui évaluent la capacité du service IoT à satisfaire les besoins des utilisateurs dans ce contexte.[19]

### 2.3 Définition de la composition des services IoT

La composition des services IoT est le processus de sélection et de combinaison de plusieurs services IoT autonomes, en tenant compte de la qualité de services, pour répondre à une demande utilisateur complexe [19]. Le résultat final de cette composition

est un nouveau service, appelé service composite [59].

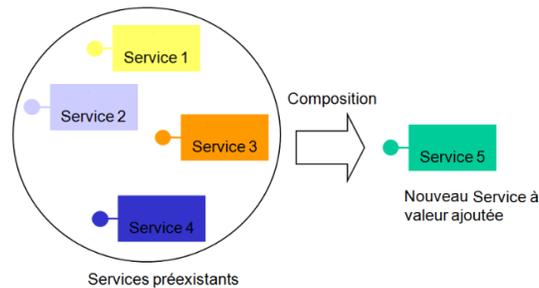


FIGURE 4 – Composition des services IoT [4].

### 2.4 Processus de la composition de services IoT tenant compte du QoS

Le processus de composition des services IoT sensible aux QoS, se déroule en quatre étapes distinctes, à savoir : l'élaboration du plan de composition, la découverte des services, la sélection de services en fonction des QoS, l'exécution de la composition.

#### 1. **Élaboration du plan de composition :**

Cette étape consiste à déterminer les exigences fonctionnelles du service composite souhaité. Les besoins de l'utilisateur sont convertis en une série de tâches fonctionnelles, organisées dans une description abstraite ou un workflow de la composition. En d'autres termes, il s'agit de décomposer la demande de l'utilisateur en un ensemble de tâches fonctionnelles qui seront ensuite intégrées dans le plan de composition.

#### 2. **Découverte des services :**

Cette phase consiste à rechercher les services IoT disponibles qui peuvent accomplir les tâches définies dans le plan de composition. Elle permet de recenser tous les services candidats fonctionnellement équivalents, regroupés dans des catégories appelées services abstraits.

#### 3. **Sélection des services en fonction de la qualité de services :**

La phase de sélection des services dans le cadre de la composition vise à identifier le service approprié pour chaque tâche abstraite, en fonction des critères de Qualité de Service. Une fois les services adaptés sélectionnés pour chaque tâche abstraite, une composition candidate est créée. Notre approche se concentre sur la problématique

## Chapitre 2 La sélection de services IoT sensible aux QoS pendant le processus de composition

de la sélection des services lors du processus de composition, en prenant en compte les critères de QoS.

### 4. Exécution et contrôle de la composition :

Pendant cette phase, la composition concrète est exécutée par un moteur d'exécution qui appelle les services composants, garantissant que la composition respecte les critères de QoS établis tout au long du processus ainsi que le changement d'un service en cas d'échec.

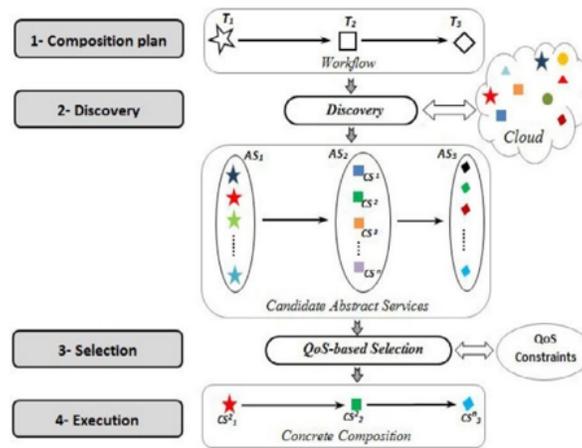


FIGURE 5 – Processus de la composition des services IoT en fonction de la QoS [11]

## 2.5 Concepts de base liés à la composition des services IoT sensibles aux QoS

### 2.5.1 Service candidat (CS)

Un service candidat, également appelé un service concret, est un service élémentaire, potentiel, invocable, qui peut être sélectionné et intégré dans une composition. En d'autres termes, il s'agit d'une instance concrète d'un service abstrait [11].

### 2.5.2 Service abstrait (AS)

Un service abstrait est une classe de services concrets partageant la même fonctionnalité, mais se distinguant par leurs attributs de QoS. En d'autres termes, il représente un modèle ou un patron pour un ensemble de services concrets partageant des fonctionnalités similaires. Ce modèle abstrait permet de définir les caractéristiques fonctionnelles essentielles d'un groupe de services, tout en laissant de la flexibilité pour la variation des attributs de qualité de service entre les différentes instances de services concrets.

### 2.5.3 Services composites abstraits (ACS)

Les services composites abstraits constituent un groupe de ( $m$ ) services abstraits interconnecter, Ils sont liés les uns aux autres par différentes structures de composition de services telles que séquentielle, parallèle, conditionnelle, en boucle ou une combinaison de celles-ci. L'ACS est conçu en fonction des besoins fonctionnels de l'utilisateur en utilisant un ensemble de services abstraits connectés [11].

### 2.5.4 Service composite concret (CCS)

Un service composite concret représente la mise en œuvre pratique d'un service composite abstrait. Cette mise en œuvre se fait en invoquant pour chaque service abstrait  $AS_i$  le service concret approprié  $CS_i$  [11].

### 2.5.5 Critères non fonctionnels (QoS)

Les critères non fonctionnels, également appelés critères de qualité de service, sont des attributs déterminant les performances, la convivialité et l'efficacité d'un système ou service, sans être directement liés à des fonctionnalités spécifiques. Il existe deux types d'attributs de QoS : les attributs négatifs (coût, temps de réponse, etc.) qui sont à minimiser et les attributs positifs, tels que la disponibilité, la fiabilité, etc qui doivent être maximiser. Voici quelques exemples de QoS avec leur définitions.

- Fiabilité : La probabilité que le service fonctionne sans défaillance pendant une période de temps spécifiée.
- Disponibilité : Le pourcentage de temps pendant lequel le service est opérationnel et accessible.
- Performance : Mesures de la rapidité du service, comme le temps de réponse ou le débit.
- Sécurité : Les mécanismes de protection des données et des transactions contre les accès non autorisés.
- Évolutivité : La capacité du service à maintenir des performances acceptables lorsqu'il est soumis à une charge accrue.
- Maintenabilité : La facilité avec laquelle le service peut être modifié ou réparé.

### 2.5.6 Vecteur de QoS d'un service concret $QoS(CS_i)$

Le vecteur des attributs de qualité de service d'un service concret, noté  $QoS(CS_i)$ , est un ensemble de valeurs ou de mesures qui caractérisent la capacité du service à répondre

## Chapitre 2 La sélection de services IoT sensible aux QoS pendant le processus de composition

de manière adéquate aux exigences de l'utilisateur. Ce vecteur inclut diverses propriétés non fonctionnelles telles que la performance, la fiabilité et la disponibilité.

Formellement, le vecteur des attributs de QoS d'un service  $CS_i$  est défini par :

$$QoS(CS_i^k) = (qos_{i1}^k, \dots, qos_{ip}^k, \dots, qos_{iq}^k),$$

où  $i \in [1..N]$ ,  $k \in [1..n]$ ,  $p \in [1..q]$ ,  $N$  est le nombre de services abstraits dans le répertoire,  $n$  est le nombre de services concrets dans le service abstrait  $AS_i$  et  $q$  est le nombre d'attributs de QoS. Le  $qos_{ip}^k$  est la valeur du  $p$ -ème attribut de QoS du  $k$ -ème service concret qui appartient au  $i$ -ème service abstrait [11].

### 2.5.7 Vecteur d'attributs QoS d'une composition $V_{QoS}(comp)$

Le vecteur d'attributs QoS d'une composition est défini par  $V_{QoS}(comp) = (Q_1, \dots, Q_q)$ , où le paramètre  $q$  représente le nombre d'attributs de qualité de service, tandis que  $Q_p$  désigne la valeur du  $p$ -ième attribut après agrégation. La manière dont les attributs sont agrégés dépend de la fonction sélectionnée (somme, produit, maximum ou minimum), comme décrit dans le tableau suivant.

Fonction	Formule	Type d'attribut
Somme	$QoS_{total} = \sum_{i=1}^n QoS_i$	Latence, Temps de réponse
Moyenne	$QoS_{moyenne} = \frac{1}{n} \sum_{i=1}^n QoS_i$	Disponibilité, Crédibilité
Produit	$QoS_{total} = \prod_{i=1}^n QoS_i$	Fiabilité
Minimum	$QoS_{min} = \min_{i=1}^n QoS_i$	Sécurité
Maximum	$QoS_{max} = \max_{i=1}^n QoS_i$	Bande passante

TABLE 1 – Fonctions d'agrégation des QoS pour le calcul du  $V_{QoS}(comp)$  [44]

## 2.6 Approches de sélection des services sensible aux QoS

L'objectif de la sélection de services avec prise en compte de QoS est de retourner une composition qui répond aux besoins de l'utilisateur.

Deux approches de sélection de services existent dans la littérature : la sélection locale et la sélection globale [3].

### 2.6.1 Sélection locale

Ces approches permettent de sélectionner, de manière individuelle, le service concret optimal en termes de qualité de service pour chaque service abstrait du plan de composition. Cependant, cette approche ne garantit pas que le service composé répondra aux exigences globales de QoS définies par l'utilisateur [11].

### 2.6.2 Sélection globale

Dans ces méthodes, les contraintes de qualité de service sont prises en considération au niveau du service composite. Une approche évidente consiste à évaluer toutes les combinaisons possibles de services, puis à identifier la meilleure, celle qui est la plus optimale en termes de QoS, tout en respectant les contraintes globales établies. Toutefois, cette méthode de sélection peut entraîner un problème de NP-hard, rendant la résolution difficile [11].

## 2.7 Approches et méthodes existantes

Pour surmonter les défis complexes de la sélection de services, considérée comme un problème NP-difficile comme mentionné dans la section précédente, des méthodes méta-heuristiques peuvent être employées pour trouver des solutions proches de l'optimale. Parmi les méta-heuristiques utilisées dans ce type d'approches, nous citons, les algorithmes génétiques, l'algorithme des colonies de fourmis, l'algorithme des abeilles. Certains travaux de la littérature choisissent de combiner des méta-heuristiques à des outils de l'intelligence artificielles comme les réseaux de neurones et les arbres de décision. Ces techniques permettent de résoudre efficacement des problèmes complexes en explorant de nombreuses solutions possibles de manière intelligente et adaptative. Dans la suite de cette section, nous fournissons plus de détails sur les approches basées sur des méta-heuristiques, puis nous définirons un outil de l'intelligence artificielle à savoir, l'apprentissage automatique.

### 2.7.1 Approches basées sur les méta-heuristique

Le terme "méta-heuristique" vient des mots grecs "meta" (au-delà) et "heuriskein" (trouver)[31].

Selon [43] dans leur article "Meta-heuristics : a bibliography", 'une méta-heuristique est formellement définie comme un processus itératif de génération qui guide une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et

## Chapitre 2 La sélection de services IoT sensible aux QoS pendant le processus de composition

---

exploiter l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions quasi-optimales'. Donc, l'objectif principal des méthodes de méta-heuristiques est d'explorer l'espace de recherche de manière efficace pour identifier des solutions qui sont proches de l'optimalité . Il y a diverses approches de méta-heuristiques comprenant une variété d'algorithmes comme le recuit simulé, l'optimisation des colonies de fourmis, l'optimisation de l'essaimage de particules, les algorithmes génétiques, et bien d'autres encore.

- **L'optimisation de la Colonie de Fourmis** : L'algorithme de la colonie de fourmis puise son inspiration à la fois dans l'évolution et dans le comportement des fourmis lorsqu'elles cherchent de la nourriture. Les fourmis démontrent une capacité remarquable à résoudre des défis complexes, tels que la recherche du chemin le plus court entre deux points dans des terrains difficiles. Leur méthode repose sur une communication locale et indirecte, facilitée par une substance chimique volatile appelée phéromone. En avançant, les fourmis laissent derrière elles une trace de phéromone, puis choisissent leur direction de manière aléatoire, basée sur la quantité de phéromone déposée [1].
- **Les algorithmes génétique** : Les algorithmes génétiques est une méthode inspirée par la nature pour explorer l'espace de recherche, les algorithmes génétiques exploitent des opérations telles que la mutation, la reproduction et la sélection pour produire de nouvelles solutions à partir des solutions existantes. Contrairement à une approche basée sur une solution unique, ils travaillent à base de population de solutions , explorant ainsi efficacement l'ensemble de l'espace de recherche. Les algorithmes génétiques sont largement utilisés dans la résolution de problèmes de composition de services IoT.
- **le recuit simulé** : est une méthode inspirée d'un processus de refroidissement des métaux. Cette méthode manipule un seul point à la fois et cherchent à améliorer itérativement ce point. Elle trace une trajectoire dans l'espace des points dans le but de progresser vers des solutions [31].

### 2.7.2 Apprentissage automatique

Le Machine Learning est un domaine de l'intelligence artificielle qui permet aux systèmes informatiques d'apprendre et de s'améliorer de manière autonome grâce à l'expérience, sans nécessiter une programmation explicite pour chaque tâche. Les algorithmes de ML utilisent des méthodes statistiques pour détecter des modèles dans les données et effectuer des prédictions ou des décisions basées sur ces informations.

## Chapitre 2 La sélection de services IoT sensible aux QoS pendant le processus de composition

---

Dans le contexte de la sélection des services IoT, ces techniques trouvent une application particulièrement pertinente. Lors de la résolution du défi de la composition de services sensible à la Qualité de Service, une approche courante est l'utilisation de l'apprentissage supervisé, une méthode faisant partie des techniques du Machine Learning. Ces approches sont fréquemment employées dans des contextes de recherche à grande échelle pour réduire l'ensemble des possibilités tout en préservant les meilleurs services disponibles. Les principales catégories de l'apprentissage automatique sont :

1. Apprentissage supervisé : L'apprentissage supervisé consiste à entraîner un algorithme à partir de données étiquetées, c'est-à-dire des ensembles de données d'entrée associés à des réponses correctes. Il existe deux types principaux d'apprentissage supervisé :
  - (a) Classification : Prédiction de catégories ou de classes.
  - (b) Régression : Prédiction de valeurs continues.
2. Apprentissage semi-supervisé : L'apprentissage semi-supervisé combine des données étiquetées et non étiquetées pour améliorer la précision de l'apprentissage. Cette approche est utile lorsque l'étiquetage des données est coûteux ou difficile.
3. Apprentissage non supervisé : L'apprentissage non supervisé se base sur des données non étiquetées. Son objectif est de découvrir des structures cachées ou des motifs dans les données. Voici quelques techniques d'apprentissage non supervisé :
  - (a) Clustering : Regroupement des données en clusters ou groupes similaires.
  - (b) Association : L'association est un processus qui vise à découvrir des relations ou des liens entre différentes données qui partagent certaines caractéristiques [31].
4. Apprentissage par renforcement : Un agent apprend à prendre des actions dans un environnement pour maximiser une récompense cumulative. Il utilise des essais et des erreurs et reçoit des récompenses ou des pénalités en fonction de ses actions.

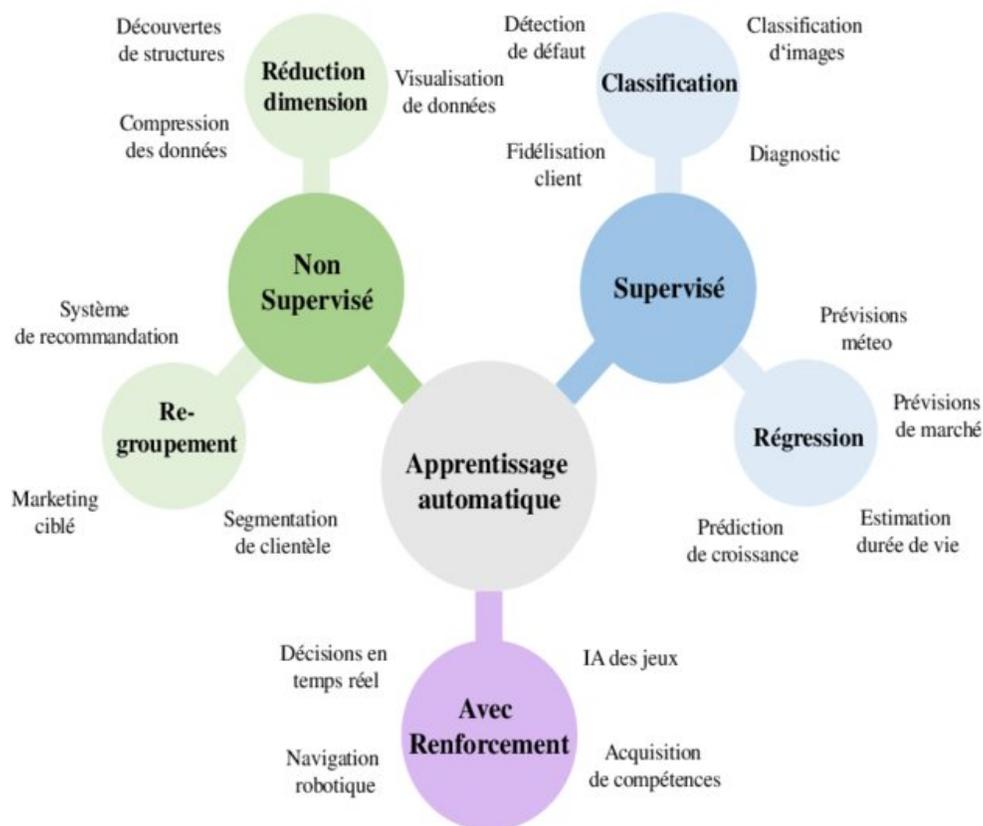


FIGURE 6 – Catégories-d'apprentissage-automatique[47].

## 2.8 Conclusion

Dans ce chapitre, nous avons défini la composition des services IoT ainsi que son processus, qui se déroule en quatre étapes : définition du plan de composition, découverte des services, sélection des services et exécution de la composition. Nous avons également introduit les concepts de base liés à la composition des services IoT. Ce mémoire traite du problème de la sélection de composition en tenant compte des critères de qualité de service et des préférences des utilisateurs. Nous avons identifié les méta-heuristiques appropriées pour résoudre ce problème, ainsi que les techniques d'apprentissage automatique pertinentes. Dans le chapitre suivant, nous explorerons plusieurs recherches sur la sélection des services IoT sensible à la QoS.

### Chapitre 3 État de l'art

#### 3.1 Introduction

Dans ce chapitre, nous étudierons plusieurs approches pour la sélection de services IoT en tenant compte des QoS. Ensuite, dans la deuxième partie, une étude comparative des approches sera traitée en se basant sur des critères tels que l'optimalité des méthodes ainsi que leur efficacité.

#### 3.2 Analyse Documentaire

Dans cette section, nous présentons les principaux travaux de recherche liés à la sélection des services IoT.

**Zebouchi et Aklouf (2024)** [60] Cet article étudie le problème de sélection des services IoT dans un environnement multiCloud. Il propose un nouvel algorithme nommé PRTMNSGA-III, qui combine les forces de RNSGA-III et TMNSGA-III. L'approche est divisée en quatre phases : prétraitement des données, détermination des tendances des mots-clés analysés, caractéristiques extraites et méthodes de classement. On utilise une évaluation parallèle pour optimiser le temps de calcul. En outre, il intègre une fonction d'appartenance floue pour sélectionner la meilleure solution en fonction des attributs de qualité de service. Les résultats obtenus des expériences démontrent que l'algorithme PRTMNSGA-III surpasse les méthodes existantes en produisant un plus grand nombre de solutions non dominées.

**Khadir et al(2022)**[27] L'article propose un algorithme évolutif appelé MOEA, utilisé pour la sélection des services IoT en tenant compte des fluctuations des paramètres de qualité de service. Elle débute par la spécification sémantique, où les besoins des applications IoT et les objets IoT hétérogènes sont modélisés sous forme d'avatars autonomes sur le Web, permettant de cerner les exigences spécifiques de l'application. La phase suivante consiste à découvrir et à identifier les avatars appropriés qui peuvent collaborer pour exécuter l'application IoT, en utilisant des algorithmes de réseautage social et de clustering pour réduire l'espace de recherche aux candidats pertinents. Ensuite, l'approche sélectionne une combinaison quasi optimale des avatars découverts en fonction de leur QoS, en décomposant les contraintes QoS globales en contraintes locales grâce à l'algorithme génétique, ce qui facilite une sélection locale efficace. Enfin, un plan d'exécution est élaboré sur la base des avatars sélectionnés, assurant que l'application IoT peut être mise

en œuvre de manière efficace selon les exigences spécifiées. Les résultats expérimentaux montrent une amélioration significative en termes de temps de calcul et d'optimalité par rapport aux méthodes traditionnelles.

**Gaurav et al(2019)[8]** L'article présente un cadre de décision de groupe multicritère (MCGDM) pour la sélection des services IoT, en se basant sur les critères de qualité de service. Ce cadre intègre les jugements linguistiques en permettant aux décideurs d'exprimer leurs évaluations avec des termes linguistiques pour différents niveaux de qualité des services en utilisant la méthode (TOPSIS flou) pour chaque DM (expert). Cette méthode identifie la meilleure alternative parmi celles disponibles. Elle ne se limite pas à trouver l'option la plus proche de la meilleure solution hypothétique, mais cherche également l'alternative la plus éloignée de la pire solution hypothétique. Les alternatives sont classées selon leur distance euclidienne entre la solution idéale et la pire solution. Ensuite, ces évaluations sont converties en nombres flous triangulaires pour faciliter la prise de décision et résoudre le problème d'inversion de rang. De plus, le cadre intègre l'opérateur OWA (Ordered Weighted Averaging Aggregation) à Fuzzy TOPSIS. Cela permet de prendre en compte l'attitude des décideurs envers le risque et d'agrèger leurs préférences pour établir un classement final. En intégrant les jugements linguistiques et les attitudes envers le risque, ce cadre offre une approche plus complète et personnalisée pour la sélection des services IoT.

**Baek et al (2023)[6]** cet article utilise DEOSA (Dynamic Effect-Oriented Service Selection Algorithm) comme une solution améliorée pour la sélection des services, en mettant l'accent sur les effets physiques dans les environnements IoT. Cet algorithme ajuste de manière dynamique la sélection et le remplacement des services de sortie afin de maximiser l'efficacité visuelle et de minimiser les coûts de remplacement. DEOSA intègre la technique d'apprentissage par renforcement de Munchausen, qui introduit des paramètres supplémentaires tels que la température d'entropie et l'alpha pour stabiliser le processus d'entraînement, garantissant ainsi un apprentissage et une prise de décision efficaces. Les résultats de la simulation montrent que DEOSA améliore significativement l'expérience utilisateur en assurant une sélection dynamique des services de sortie basée sur les effets physiques, ce qui renforce la pertinence des applications IoT.

**Karimi et Morteza (2017)[25]** Dans cet article, les auteurs étudient le problème de la sélection des meilleurs services parmi plusieurs candidats, un défi complexe en raison de la répartition des valeurs optimales des paramètres de qualité et des conflits entre ces derniers, tels que le coût et le temps de réponse. Ils proposent d'adapter l'algo-

rithme Grey Wolf Optimizer (GWO) pour sélectionner les services web candidats en vue d'obtenir une composition optimale. Cette méthode est comparée à plusieurs variantes de l'algorithme Particle Swarm Optimization (PSO). L'approche GWO montre une bonne évolutivité pour la composition de services web, tout en prenant en compte la qualité de service. Sa capacité à trouver des solutions optimales de manière efficace et sa simplicité d'implémentation en font un choix prometteur pour ce type de problème.

**Mahfoudh et al (2018)** [35] ont proposé un cadre de composition de services reposant sur un modèle de coordination avec apprentissage par renforcement. Leur approche intègre l'apprentissage par renforcement multi-agents, un modèle de coordination inspiré par la nature (utilisant des règles de coordination basées sur des réactions chimiques) et des services auto-composants.

**D'Angelo et al(2020)** [15] ont présenté une méthode d'apprentissage décentralisée pour la composition de services auto-adaptatifs sensibles à la QoS. Cette technique utilise l'apprentissage par renforcement pour permettre aux services d'apprendre de manière dynamique à partir des expériences passées, en s'appuyant sur un modèle de partage d'informations pour structurer la boucle MAPE-K décentralisée.

**Xiao et al(2022)**[58] présente une version améliorée de l'algorithme Grey Wolf Optimizer pour réaliser la composition de services web à grande échelle. Cet algorithme intègre la théorie du chaos et un facteur de convergence non linéaire dans l'algorithme Grey Wolf. Des expériences parallèles utilisant le cadre MapReduce ont été menées, démontrant une performance supérieure en termes de fitness moyenne et de stabilité par rapport à d'autres algorithmes. La méthode proposée répond efficacement aux défis de l'optimisation composite pour les services web à grande échelle.

**Tang et al(2024)** [55] L'article propose une version améliorée de l'algorithme de saut de grenouille, nommé Improved Shuffled Frog-Leaping Algorithm (ISFLA), dérivé de l'algorithme original (SFLA). L'ISFLA combine la stratégie d'apprentissage mutuel du SFLA avec l'approche évolutive des algorithmes génétiques (GA) pour améliorer l'optimisation. En intégrant la théorie du chaos et de l'apprentissage inverse, il optimise la population initiale pour une meilleure composition des services IoT basée sur la qualité de service. L'algorithme utilise la distance euclidienne pour diviser la population en groupes, facilitant la sélection et l'optimisation des individus. La mutation gaussienne optimise le meilleur individu de chaque groupe, améliorant ainsi le processus global. Une méthode de mise à jour locale de la population assure une évolution vers des solutions optimales. Les simulations montrent que l'ISFLA offre de meilleures valeurs de fitness, des solutions plus efficaces, une convergence rapide, une stabilité accrue et une meilleure

efficacité opérationnelle comparée à d'autres algorithmes.

**khanouche et al(2019)** [29] ont introduit une méthode de sélection de services basée sur le clustering avec K-means pour regrouper et éliminer les services candidats selon leur niveau de QoS. Ils ont également proposé une optimisation lexicographique pour déterminer les services répondant aux contraintes globales de QoS, utilisant un arbre de recherche pour obtenir le service composite quasi optimal.

**Khanouche et al(2016)** [28] propose un algorithme de sélection de services axé sur l'énergie et sensible à la QoS (EQSA) est proposé pour identifier de manière efficace les services les plus appropriés parmi un ensemble, tout en optimisant la consommation d'énergie. L'approche commence par une phase de pré-sélection qui examine tous les services offrant un niveau de QoS essentiel pour satisfaire les demandes de l'utilisateur. Cette étape aboutit à un ensemble de services candidats qui seront évalués par la suite. Après ces premières étapes, les services les plus adaptés sont sélectionnés en fonction de leur profil énergétique et de leurs préférences en matière de QoS. Les performances de l'algorithme EQSA ont été évaluées par des simulations approfondies, montrant des résultats prometteurs en termes de performance.

**Wang et al (2019)** [57] ont introduit une nouvelle approche avancée pour l'optimisation de la composition des services appelée MRED . Elle se distingue par sa capacité à intégrer des techniques d'estimation de distribution, des machines de Boltzmann restreintes (RBM), et des systèmes MA (multi-agents) pour identifier des services composites optimaux. MRED combine les algorithmes d'estimation de distribution (EDA) et les RBM pour modéliser des distributions complexes et capturer les interactions entre les variables. Les systèmes multi-agents coordonnent la recherche de solutions optimales par une exploration coopérative, améliorant ainsi la diversité et l'efficacité des solutions trouvées. L'optimisation avec MRED est un processus itératif qui affine continuellement les compositions de services en tenant compte des variations des services et de la qualité de service . Grâce à sa capacité d'ajustement adaptatif, MRED peut réagir aux changements dans l'environnement opérationnel, garantissant des compositions efficaces même sous des conditions dynamiques.

**Chai et al (2021)** [12] Dans cet article, les auteurs proposent une nouvelle méthode appelée FSCA-EQ (Fast Service Composition Approach Energy and Quality-aware) pour la composition de services dans l'Internet des objets. Cette approche utilise une méthode d'optimisation hiérarchique qui commence par présélectionner les services basés sur leurs performances en termes de qualité de service. Ensuite, elle effectue une sélection

tion finale en tenant compte de la consommation d'énergie et des préférences de l'utilisateur. Cette méthode permet de composer des services de manière rapide et efficace, tout en optimisant les aspects énergétiques et la qualité de service. Cependant, elle présente certains défis, tels que l'adaptabilité aux changements dynamiques des préférences des utilisateurs, la complexité de l'intégration de services hétérogènes, la gestion efficace de l'énergie, l'évaluation subjective de la QoS et les limitations de scalabilité à grande échelle.

**Kashyap et al (2020)** [26] Dans cet article, les auteurs présentent l'algorithme NSGA II (Non-Dominated Sorting Genetic Algorithm II) pour l'optimisation multi-objectifs dans la composition de services IoT. Cette approche permet d'optimiser simultanément plusieurs critères, tels que la qualité de service, le coût et l'efficacité, offrant ainsi une flexibilité et une adaptation aux besoins variés des utilisateurs. Cependant, l'utilisation de cet algorithme nécessite des ressources computationnelles élevées, en particulier pour des problèmes complexes, ce qui peut représenter un défi dans des environnements IoT à grande échelle.

### 3.3 Étude comparative entre les approches

Après avoir examiné des travaux fondés sur des techniques déterministes ainsi que sur des méta-heuristiques, nous avons constaté que les méta-heuristiques se démarquent par leur approche flexible et adaptative pour résoudre des problèmes complexes. Contrairement aux méthodes déterministes, comme la programmation linéaire, et la recherche exhaustive, qui suivent des algorithmes basés sur des règles strictes, les méta-heuristiques adoptent des stratégies plus évolutives. Les méthodes traditionnelles sont souvent efficaces pour des problèmes bien structurés et avec des espaces de recherche restreints, mais elles peuvent s'avérer inadaptées face à des défis plus complexes impliquant des espaces de recherche vastes et mal définis. En revanche, les méta-heuristiques montrent une grande adaptabilité et une capacité à explorer l'espace de recherche de manière aléatoire et itérative, leur permettant ainsi de trouver des solutions de qualité dans des contextes difficiles. Cette comparaison souligne que les méta-heuristiques excellent dans la résolution de problèmes complexes où les méthodes traditionnelles atteignent leurs limites, ce qui en fait un choix précieux dans de nombreux domaines d'application, y compris pour la sélection de services IoT lors du processus de composition en tenant compte des QoS [54].

Cependant, il existe également d'autres approches, telles que celles qui intègrent l'avis des experts lors de la sélection, ou encore des méthodes révolutionnaires qui combinent ces approches traditionnelles avec des innovations technologiques récentes. Ces méthodes hybrides, en plus de bénéficier de l'évolutivité des méta-heuristiques, apportent une dimension supplémentaire en associant l'intelligence humaine et la technologie pour affiner encore davantage la prise de décision dans des environnements complexes.

Nous concluons cette section par une étude comparative des travaux présentés ci-dessus. Les critères de comparaison des différentes approches sont présentés ci-dessous :

— **Année et Auteur**

Ce critère fait référence à l'année de publication et aux auteurs des articles, permettant ainsi de situer la recherche dans le temps et d'évaluer la pertinence des informations dans le domaine de la composition de services IoT.

— **Approche Proposée**

Il s'agit de la méta-heuristique ou bien d'autre méthode utilisée dans la composition des services IoT étudiée.

— **Optimalité**

L'optimalité fait référence à la capacité de fournir une solution optimale ou proche de l'optimum. Dans le cadre de l'optimisation, une solution est considérée comme optimale lorsqu'elle maximise ou minimise la fonction objective tout en respectant

l’ensemble des contraintes du problème, comme la minimisation de la latence ou la maximisation de la disponibilité, par exemple.

### — Avantages et Limites

Ce critère met en évidence les points forts et les faiblesses de chaque approche proposée.

Le tableau 2 présente une étude comparative des travaux antérieurs examinés dans la littérature

Article	Approche Proposée	Optimalité	Avantage	Inconvénient
Article 1 : Zebouchi et Ak-louf (2024) [60]	Méta-heuristique hybride PRTMNSGA-III, combinant RNSGA-III et TMNSGA-III	Optimale	Parallélisation pour une meilleure gestion des ressources Focus sur la qualité de service (QoS)	Complexité de conception et de mise en œuvre Augmentation potentielle du coût
Article 2 : Khadir et al. (2022) [27]	Algorithme évolutif multi-objectifs (MOEA)	Quasi-optimale	Fournit des solutions optimales plus stables dans des environnements dynamiques	Complexité de mise en œuvre
Article 3 : Gaurav et al. (2019) [8]	Cadre de prise de décision de groupe multicritères (MCGDM)	Quasi-optimale	Approche structurée pour exprimer les exigences QoS	Efficacité variable selon le contexte spécifique
Article 4 : Baek et al. (2022) [6]	DEOSA (Dynamic Effect-Oriented Service Selection Algorithm)	Optimale	Flexibilité accrue pour gérer un nombre variable de services candidats	Difficulté à comprendre et à gérer efficacement les pénalités de remplacement

Article	Approche Proposée	Évolutivité	Avantage	Inconvénient
Article 5 : Karimi et Mor- teza (2017) [25]	Algorithme Grey Wolf Optimizer	Optimale	Efficacité dans la gestion de la com- plexité de la com- position des ser- vices	La méthode peut ne retenir qu'une seule solution optimale, excluant d'autres options similaires
Article 6 : Mahfoudh et al. (2018) [35]	Cadre de com- position de ser- vices avec ap- prentissage par renforcement	Optimal	Adaptation en temps réel et qualité de service fiable et cohérente	Complexité de mise en œuvre
Article 7 : D'Angelo et al. (2020) [15]	Méthode d'ap- prentissage dé- centralisée	Optimal	L'apprentissage par renforcement permet au sys- tème d'améliorer continuellement ses règles d'as- semblage basées sur les expériences passées	Dépendance à l'apprentissage
Article 8 : Xiao et al. (2022) [58]	Version amé- liorée de l'al- gorithme Grey Wolf Optimizer (GWO)	Quasi- optimale	Performance supérieure en termes de fitness moyenne	Complexité d'im- plémentation
Article 9 : Tang et al. (2024) [55]	Algorithme amélioré de saut de gre- nouille mélangé (ISFLA)	Quasi- optimale	Accélère la conver- gence et améliore l'efficacité opéra- tionnelle	Complexité accrue par rapport à des algorithmes plus simples

Article	Approche Proposée	Évolutivité	Avantage	Inconvénient
Article 10 : Khanouche et al. (2019) [29]	Méthode de sélection basée sur le clustering K-means	Quasi-optimale	Réduit efficacement l'espace de recherche en utilisant le clustering, ce qui améliore les performances globales	Le recours au clustering peut introduire des erreurs susceptibles d'affecter la composition finale
Article 11 : Khanouche et al. (2016) [28]	Energy-centered and QoS-aware Services Selection Algorithm (EQSA)	Quasi-optimale	Réduit la consommation d'énergie tout en répondant aux exigences de QoS	Sensibilité aux paramètres
Article 12 : Wang et al. (2019) [57]	MRED (Algorithme d'estimation réactive multi-agents de distribution)	Quasi-optimale	Permet une adaptabilité accrue	Complexité de la mise en œuvre
Article 13 : Kashyap et al. (2020) [26]	Algorithme NSGA II (Non-Dominated Sorting Genetic Algorithm II)	Optimale	Optimisation multi-objectifs pour la composition de services, permettant une flexibilité et une adaptation aux besoins variés	Nécessite des ressources computationnelles élevées pour des problèmes complexes

Article	Approche Proposée	Évolutivité	Avantage	Inconvénient
Article 14 : Chai et al. (2021) [12]	FSCA-EQ (Fast Service Composition Approach Energy and Quality-aware)	Optimale	Composition rapide et efficace des services en optimisant la consommation d'énergie et la qualité de service	Adaptabilité limitée aux changements dynamiques des préférences des utilisateurs, complexité d'intégration des services hétérogènes, gestion de l'énergie, évaluation subjective de la QoS, limitations de scalabilité
Approche proposée KM-GWO	K-means avec Grey Wolf Optimization (GWO)	Optimale	Réduction de l'espace de recherche et retour du meilleur service composite en optimisant la QoS	GWO peut écarter des solutions alternatives mieux adaptées aux besoins des utilisateurs en ne privilégiant que le service IoT avec la meilleure valeur de fitness

TABLE 2 – Comparaison des méthodes de sélection de services IoT

## Chapitre 3 État de l'art

---

Dans le tableau suivant 3, nous mentionnons les critères de QoS traités dans chaque article, tels que le temps d'exécution, le coût des services, la crédibilité et la fiabilité. Chaque approche est évaluée selon un cadre standardisé. Pour indiquer le degré de satisfaction des critères QoS, nous utilisons des symboles spécifiques : (+) pour satisfait, (-) pour insatisfait, (/) pour non mentionné, et (N/A) pour non disponible.

Article	Temps d'exécution	Coût des services	Crédibilité	Fiabilité
Article 1 : [60]	+	+	+	+
Article 2 : [27]	+	+	+	+
Article 3 : [8]	+	+	+	-
Article 4 : [6]	-	-	-	-
Article 5 : [25]	+	+	+	+
Article 6 : [58]	+	+	+	+
Article 7 : [55]	+	+	+	+
Article 8 : [29]	+	+	+	-
Article 9 : [15]	+	+	+	-
Article 10 : [35]	+	+	+	-
Article 11 : [28]	+	+	+	-
Article 12 : [57]	+	+	+	+
Article 13 : [26]	+	+	+	+
Article 14 : [12]	+	+	+	-

Article	Temps d’exécution	Coût des services	Crédibilité	Fiabilité
Notre travail(2024)	+	+	+	+

TABLE 3 – Critères traités dans chaque article

### 3.4 Discussion

La Table 3 montre que la plupart des approches récentes traitant le problème de composition de services IoT sensible aux QoS se concentrent sur l’optimisation du temps d’exécution et du coût des services. Les études, notamment celles de Zebouchi et al. (2024) [60], Khadir et al. (2022) [27], Karimi et al. (2017) [25], Xiao et al. (2022) [58], et Tang et al. (2024) [55], confirment cette tendance en cherchant à améliorer l’efficacité et la rentabilité. Ces méthodes privilégient souvent l’utilisation d’approches non déterministes, comme les algorithmes méta-heuristiques, pour explorer de vastes espaces de recherche et fournir des solutions de haute qualité dans des délais raisonnables.

L’approche FSCA-EQ de Chai et al. (2021) [12] se distingue par son optimisation de la consommation d’énergie tout en maintenant la qualité de service, un aspect crucial dans les environnements IoT. Cependant, cette méthode doit faire face à des défis, notamment l’adaptabilité aux préférences changeantes des utilisateurs et la complexité d’intégration de services hétérogènes.

La crédibilité est un critère généralement bien traité par toutes les approches examinées. En revanche, la fiabilité des services n’est pas toujours explicitement abordée, comme le soulignent les travaux de Gaurav et al. (2019) [8] et Khanouche et al. (2019) [29]. Les approches de Zebouchi et al. (2024) [60] et Kashyap et al. (2020) [26] sont particulièrement complètes, satisfaisant tous les critères QoS (+/+ /+ /+), ce qui les rend adaptées aux environnements IoT exigeants.

Certaines méthodes, comme celles de Khadir et al. (2022) [27] et D’Angelo et al. (2020) [15], omettent de traiter la fiabilité, ce qui peut poser des problèmes dans des contextes critiques. Les approches déterministes, telles que celles de Khanouche et al. (2019) [29] et Khanouche et al. (2016) [28], offrent des solutions garanties, mais peuvent devenir inefficaces pour des problèmes complexes en raison de l’explosion combinatoire, souvent en raison de la nature NP-dure de ces problèmes.

Les approches non déterministes, utilisant des algorithmes méta-heuristiques comme dans Karimi et al. (2017) [25], Khadir et al. (2022) [27], Tang et al. (2024) [55], Zebouchi et al. (2024) [60], Xiao et al. (2022) [58], Kashyap et al. (2020) [26], et Chai et al. (2021) [12],

permettent une gestion plus flexible de la complexité en explorant l'espace de recherche de manière plus adaptative.

Enfin, la réduction de l'espace de recherche, comme démontré dans les études Karimi et al. (2017) [25] et Khanouche et al. (2019) [29], est une stratégie efficace pour améliorer l'efficacité du processus de sélection des services, tout en réduisant la complexité du problème.

### 3.5 Conclusion

Dans ce travail, nous avons abordé le problème de la sélection de la composition en tenant compte des critères de qualité de service et des préférences des utilisateurs. Nous avons constaté que, bien que la crédibilité soit généralement bien traitée par les différentes approches examinées, la fiabilité des services reste un aspect souvent négligé, ce qui pourrait poser des problèmes dans des contextes critiques. Les travaux de [60] et [26] se distinguent par leur exhaustivité, satisfaisant tous les critères QoS, ce qui les rend particulièrement adaptés aux environnements IoT exigeants. En conclusion, notre étude souligne l'importance de l'intégration de techniques de clustering et d'optimisation pour améliorer la composition des services IoT. Les résultats obtenus ouvrent la voie à des recherches futures visant à affiner ces méthodes et à explorer de nouvelles approches pour garantir la fiabilité et l'adaptabilité des services dans un environnement en constante évolution.

### Chapitre 4 : Approche proposée : K-means with Grey Wolf Optimization Algorithm

#### 4.1 Introduction

La composition de services permet de répondre à des demandes complexes en combinant plusieurs services pour créer des services composites à valeur ajoutée et réutilisables. Le principal défi réside dans la satisfaction des besoins des utilisateurs en prenant en compte divers critères de QoS, à maximiser et à minimiser, dans un vaste espace de recherche. Cette complexité rend le problème de composition de services, sensible aux QoS, NP-difficile. Dans ce chapitre, nous présentons notre approche, K-means with Grey Wolf Optimization Algorithm (KM-GWO), qui associe l'algorithme de clustering K-means à l'algorithme Grey Wolf Optimizer pour résoudre le problème de composition de services IoT en tenant compte des critères de QoS.

#### 4.2 Motivation

Dans le but de surmonter la complexité du problème de composition de services IoT et d'optimiser à la fois l'efficacité et la qualité, nous avons choisi l'approche K-means with Grey Wolf Optimization Algorithm. Cette approche combine deux techniques pour répondre aux défis liés à la gestion de l'espace de recherche vaste et à la satisfaction des préférences de QoS de l'utilisateur en retournant une solution quasi-optimale. L'algorithme K-means est d'abord utilisé pour classer les services IoT selon leurs attributs de QoS, ce qui permet de réduire l'espace de recherche en regroupant les services similaires. Cette réduction simplifie le processus de sélection en réduisant le nombre de combinaisons à explorer et en supprimant les services candidats les moins pertinents pour l'utilisateur. Ensuite, l'algorithme GWO est appliqué pour rechercher une composition quasi-optimale dans cet espace réduit. Grâce à sa capacité à équilibrer l'exploration et l'exploitation de l'espace de recherche, GWO identifie des solutions qui garantissent des temps d'exécution raisonnables tout en maintenant un niveau élevé de qualité de service.

#### 4.3 L'algorithme K-Means

K-means clustering est un algorithme d'apprentissage automatique non supervisé qui regroupe un ensemble de données non étiquetées en différents clusters. L'algorithme K-Means, également connu sous le nom d'algorithme de partitionnement par centre mobile, permet de diviser un ensemble de données en K clusters. Chaque cluster représente un

regroupement de concepts similaires, et son centre est dynamique tout au long de l'exécution de l'algorithme [23]. Le processus de l'algorithme peut être décrit comme suit :

- Le nombre de clusters, défini par le paramètre  $K$ , est initialement spécifié.
- Un ensemble de  $K$  centres est sélectionné parmi les données disponibles.
- Les  $K$  clusters sont formés en regroupant les données les plus proches du centre actuel dans chaque cluster.
- Le centre de chaque cluster est recalculé et devient le nouveau centre.
- L'algorithme répète ces étapes jusqu'à ce que les centres deviennent immobiles, indiquant ainsi la condition d'arrêt du processus.

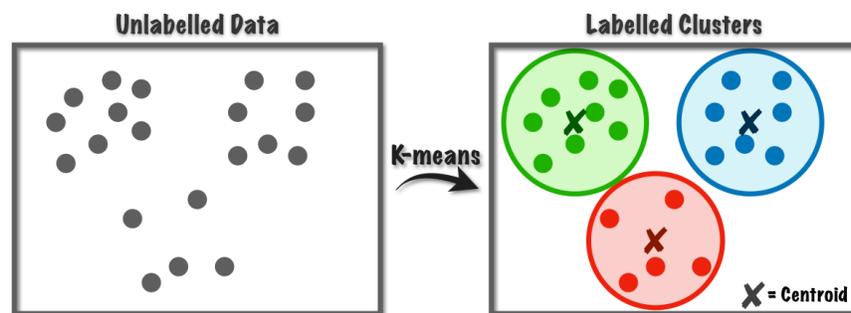


FIGURE 7 – K-means[23]

### 4.4 Grey Wolf Optimization (Optimisation par Loup Gris)

GWO [38] a été introduit en 2014 par Seyedali Mirjalili et al. et s'est rapidement imposée comme un outil puissant pour résoudre des problèmes d'optimisation complexes dans divers domaines. Cette méta-heuristique s'inspire de la structure sociale et des techniques de chasse des loups gris, qui possèdent une hiérarchie sociale clairement définie, comme illustré dans la Figure 8. Dans cette hiérarchie, les loups alpha, bêta et delta occupent des positions de leadership, tandis que les loups oméga se trouvent au bas de l'échelle. L'algorithme GWO utilise cette hiérarchie pour orienter le processus de recherche et guider l'optimisation.

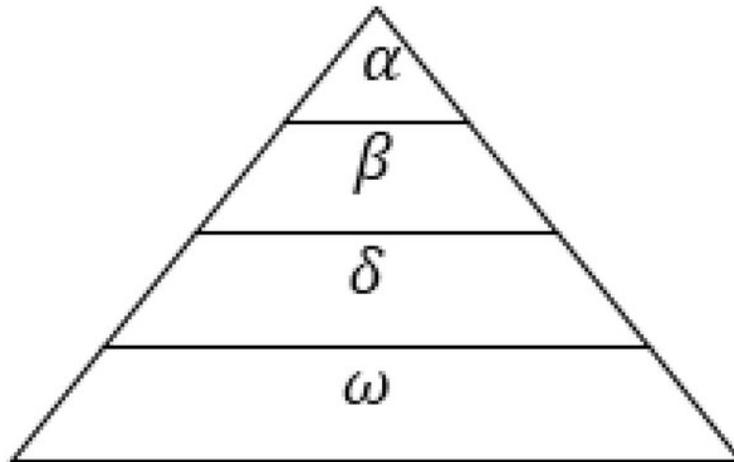


FIGURE 8 – Hierarchy of grey wolf.[13]

Dans les sections suivantes, nous allons détailler les opérateurs du GWO, à savoir :

### 4.4.1 La recherche de proies

L'algorithme d'optimisation des loups gris s'inspire du comportement de chasse des loups gris, en s'appuyant sur les rôles des loups alphas, bêtas et deltas pour guider la meute. Ces loups commencent par se disperser pour explorer l'espace de recherche, identifiant les meilleures solutions potentielles, avant de converger vers des solutions pour les exploiter plus en profondeur. Le paramètre  $A$  est essentiel dans cette dynamique, régulant la divergence et la convergence des loups. Le vecteur de coefficients  $A(t)$ , utilisé pour réguler la divergence et la convergence des loups, est calculé selon la formule 4. Si  $A > 1$ , les loups s'éloignent pour explorer davantage, alors que si  $A < 1$ , ils se rapprochent des solutions prometteuses. Le paramètre  $a$ , qui diminue linéairement de 2 à 0, ajuste  $A$  au fil des itérations, assurant une transition équilibrée entre exploration et exploitation. En parallèle, le paramètre  $C$ , avec des valeurs aléatoires comprises entre 0 et 2, influence l'attraction des loups vers les proies. Quand  $C < 1$ , l'exploration est favorisée en réduisant l'attraction vers les solutions actuelles, permettant d'éviter les optima locaux.  $C(t)$ , utilisé pour ajuster l'attraction des loups, est défini par la formule 2. Ces mécanismes combinés permettent au GWO de rechercher efficacement les solutions optimales en équilibrant la diversité de l'exploration et la précision de l'exploitation.

### 4.4.2 L'encerclement de la proie

Le mécanisme d'encerclement de la proie par les loups gris dans GWO se déroule en deux phases principales : l'exploration globale et l'exploitation locale. Les loups se dispersent d'abord pour explorer l'espace de recherche et identifier les meilleures solutions

potentielles. Ensuite, ils convergent pour attaquer ces solutions, ce qui correspond à l'exploitation locale.

Pour modéliser mathématiquement le comportement d'encerclement, les équations suivantes sont proposées :

La distance  $\mathbf{D}(t)$  entre un loup gris et la proie est calculée par :

$$\mathbf{D}(t) = |\mathbf{C}(t) \times \mathbf{X}_p(t) - \mathbf{X}_i(t)| \quad (1)$$

où  $\mathbf{X}_i(t)$  c'est la position d'un loup spécifique à l'itération  $t$  (searche agent) ,  $\mathbf{X}_p(t)$  c'est la position de la proie , et  $\mathbf{C}(t)$  est un vecteur de coefficients défini comme :

$$\mathbf{C}(t) = 2 \times \mathbf{r}_2 \quad (2)$$

avec  $\mathbf{r}_2$  étant un vecteur aléatoire compris entre 0 et 1.

La position du loup gris à l'itération  $t + 1$  est mise à jour par :

$$\mathbf{X}_i(t + 1) = \mathbf{X}_p(t) - \mathbf{A}(t) \times \mathbf{D}(t) \quad (3)$$

où  $\mathbf{A}(t)$  est un vecteur de coefficients calculé comme :

$$\mathbf{A}(t) = 2 \times \mathbf{a} \times \mathbf{r}_1 - \mathbf{a} \quad (4)$$

avec  $\mathbf{a}$  étant un vecteur qui diminue linéairement de 2 à 0 au cours des itérations, et  $\mathbf{r}_1$  un vecteur aléatoire compris entre 0 et 1.

En réduisant linéairement le vecteur  $\mathbf{a}$  au fil des itérations, ce mécanisme favorise une transition équilibrée entre l'exploration initiale et l'exploitation finale. Cela permet aux loups d'encercler et d'attaquer efficacement la proie, en évitant de tomber dans des optima locaux.

### 4.4.3 La chasse de la proie

Dans GWO, les positions d'alpha, bêta , et delta sont toujours les trois meilleures solutions obtenues. Il est supposé que ces loups ont une meilleure connaissance de l'emplacement potentiel des proies. Par conséquent, les autres loups doivent mettre à jour leurs positions en fonction des positions des loups alpha, bêta et delta. Le modèle mathématique de la chasse est défini par les équations suivantes [41] :

**Calcul des vecteurs de distance**

$$\mathbf{D}_\alpha = |\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}_i(t)| \quad (5)$$

$$D_{\beta} = |C_2 \cdot X_{\beta} - X_i(t)| \quad (6)$$

$$D_{\delta} = |C_3 \cdot X_{\delta} - X_i(t)| \quad (7)$$

Mise à jour des positions des trois meilleurs loups :

$$X_1 = X_{\alpha} - A_1 \cdot D_{\alpha} \quad (8)$$

$$X_2 = X_{\beta} - A_2 \cdot D_{\beta} \quad (9)$$

$$X_3 = X_{\delta} - A_3 \cdot D_{\delta} \quad (10)$$

Mise à jour de la position des autres loups

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (11)$$

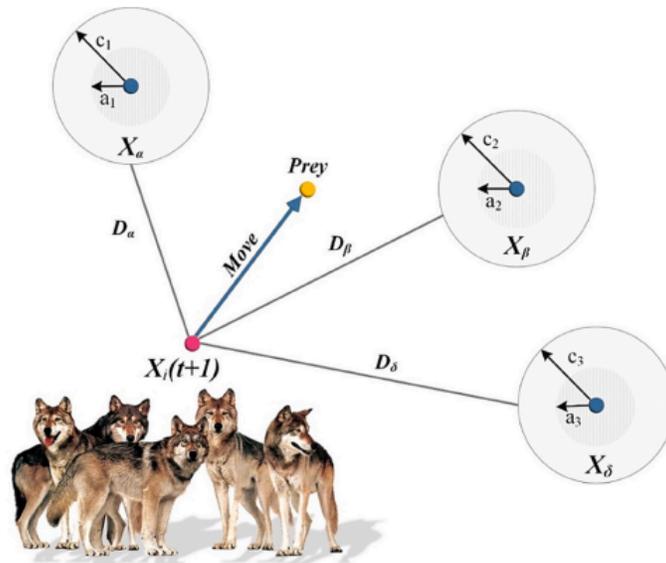


FIGURE 9 – Mise à jour de position par les loups gris.[41]

#### 4.4.4 L'attaque de proie

L'attaque de la proie dans GWO est une stratégie essentielle où les loups gris terminent la chasse en attaquant la proie. Cette phase, connue sous le nom de stratégie d'exploita-

tion, est activée lorsque la valeur absolue du paramètre  $A$  est inférieure à un. La variable  $A$  est générée aléatoirement dans l'intervalle  $[-2a, 2a]$ , où  $a$  diminue progressivement de 2 à 0 au fil des itérations. Cette technique assure une recherche locale intense tout en maintenant un équilibre optimal entre exploration et exploitation grâce au paramètre  $a$ , qui est mis à jour selon la formule :

$$a = 2 - \frac{2t}{T_{\max}} \quad (12)$$

où  $T_{\max}$  représente le nombre maximum d'itérations.

Dans la figure suivante 10, l'algorithme GWO est présenté en détail. Cet algorithme simule le comportement de chasse des loups gris en modélisant les interactions entre les loups alpha, bêta et delta, qui représentent les meilleures solutions potentielles. Les autres loups (agents) ajustent leurs positions en fonction de ces solutions dominantes. Cette approche itérative permet de guider efficacement la recherche vers l'optimum global en évitant les pièges des optima locaux. Le processus est décrit mathématiquement par les équations de mise à jour des positions et d'encerclement, comme illustré dans les sections précédentes. L'algorithme est conçu pour exploiter au maximum la synergie entre les capacités exploratoires et exploitatives des loups, assurant ainsi une convergence rapide et précise vers la meilleure solution.

Input	Population size is $N$ ; maximum number of iterations is $T$ ;current iteration $t=1$ ; population dimension is $d$
Output	Optimal solution $X_\alpha$
1.	Randomly initialize the gray wolf population $X_i(i=1,2,\dots,n)$ and associated parameters $a$ , $A$ and $C$ ;
2.	Calculate the fitness value of each individual gray wolf;
3.	Obtain the current top three gray wolf individuals in terms of fitness and define them as $X_\alpha$ , $X_\beta$ and $X_\delta$ respectively;
4.	<b>while</b> ( $t < T$ )
5.	for each individual gray wolf position is updated using Equation (7);
6.	update the relevant parameters $a$ , $A$ and $C$ ;
7.	calculating the fitness of each individual gray wolf;
8.	updating $X_\alpha$ , $X_\beta$ and $X_\delta$ ;
9.	$t = t + 1$
10.	<b>end while</b>
11.	Return $X_\alpha$

FIGURE 10 – L'algorithme GWO original [48]

### 4.5 Fonctionnement de l'approche proposée KM-GWO

L'approche KM-GWO combine l'algorithme de clustering K-means et l'algorithme GWO pour résoudre le problème de composition de services IoT en prenant en compte les

critères de qualité de service . Dans KM-GWO, l'algorithme K-means est utilisé pour réduire l'espace de recherche en classant les services IoT selon leurs attributs QoS. Ensuite, l'algorithme GWO est appliqué pour trouver une solution quasi-optimale qui répond aux besoins des utilisateurs en termes de QoS.

Dans cette section, nous expliquons la mise en œuvre de notre approche. Nous présentons les deux étapes principales de KM-GWO, à savoir la classification des services IoT à l'aide de l'algorithme K-means et la sélection d'une composition quasi-optimale des services en utilisant l'algorithme GWO.

### 4.5.1 Classification des services candidats via K-means

#### La normalisation des valeurs des QoS du dataset

Avant d'entamer la classification des services, il est essentiel de normaliser leurs valeurs de QoS pour les ramener dans l'intervalle  $[0,1]$  [49]. Pour ce faire, nous avons utilisé les formules suivantes pour normaliser les valeurs des attributs de QoS [19]

#### — Attributs négatifs

$$q'_{(i,j)} = \begin{cases} \frac{q_j^{\max} - q_{(i,j)}}{q_j^{\max} - q_j^{\min}} & \text{si } q_j^{\max} - q_j^{\min} \neq 0 \\ 1 & \text{sinon} \end{cases} \quad (13)$$

#### — Attributs positifs

$$q'_{(i,j)} = \begin{cases} \frac{q_{(i,j)} - q_j^{\min}}{q_j^{\max} - q_j^{\min}} & \text{si } q_j^{\max} - q_j^{\min} \neq 0 \\ 1 & \text{sinon} \end{cases} \quad (14)$$

Où  $q'_{(i,j)}$  représente la valeur normalisée de l'attribut  $j$  de la QoS associée au service candidat  $SC_i$ . Elle est calculée en utilisant la valeur actuelle  $q_{i;j}$ , ainsi que les valeurs maximales  $q_{\max;j}$  et minimales  $q_{\min;j}$  de l'attribut  $j$  de la QoS.

#### La classification des services IoT

Après avoir préparé et entraîné le modèle K-means, nous l'utilisons pour classer et regrouper les services du dataset IoT en clusters. Nous avons initialement défini trois clusters, représentant des groupes de services abstraits  $AS_i$ . Ensuite, chaque service candidat est assigné à l'un des clusters, identifiés par les couleurs verts, violet et jaune.

L'algorithme 1 décrit les étapes du processus de classification des services à l'aide de K-means.

---

**Algorithm 1:** Algorithme K-means

---

**Input:** Dataset IoT, poids, num-clusters

**Output:** services IoT normalisé et classé, regroupé dans chaque Clusters

**1 Début**

2 Chargement des données.

3 Prétraitement des données.

4 Lire le Dataset IoT.

5 Normalisé les données de Dataset IoT.

6 Application de KMeans.

7 Évaluation des clusters.

**8 Fin**

---

### 4.5.2 Application de GWO

Nous représentons notre modélisation de la problématique de composition de services IoT en prenant en compte les critères de Qualité de Service. Une composition abstraite, ou un plan de composition, notée SCA, se compose de  $n_{AS}$  services abstraits (AS) enchaînés, représentés comme suit :  $SCA = \{AS_1, AS_2, \dots, AS_{n_{AS}}\}$ . Chaque AS contient un ensemble de  $n_{CS}$  services candidats fonctionnellement équivalents, définis par  $AS = \{CS_1, CS_2, \dots, CS_{n_{CS}}\}$ . Une composition est représentée par un vecteur entier de taille  $n_{AS}$ , exprimé comme suit :  $comp = \{x_1, x_2, \dots, x_{n_{AS}}\}$ , où  $x_i$  représente l'indice du service candidat choisi pour le  $i$ -ème service abstrait. Dans la suite de cette section, nous présenterons en détail les étapes de l'application de GWO.

— **Initialisation de la population**

Dans cette étape, GWO sélectionne aléatoirement les services candidats  $CS_i$  dans chaque cluster afin de former des services composites.

— **Génération aléatoire des compositions de la population initiale**

À cette étape, des compositions sont générées aléatoirement à travers l'espace de recherche. Dans chaque composition, les services  $CS_i$  doivent rester dans les limites du nombre de services candidats disponibles dans chaque AS, ce qui signifie que les  $x_i$  de chaque composition doivent être compris entre 1 et  $n_{CS}$ .

— **Calcul des vecteurs de qualité de service des compositions générées  $V_{QoS(Comp_i)}$**

Après avoir initialisé la population, nous avons regroupé les valeurs de QoS pour

analyser les compositions de services IoT. Cette étape consiste à calculer des mesures globales pour chaque composition, en tenant compte des différents attributs de QoS . Les équations de calcul appropriées sont ensuite appliquées pour obtenir les valeurs finales de QoS.

— **Temps d'exécution**

$$Q_{te} = \sum_{i=1}^n q_{te_i} \quad (15)$$

— **Coût du service**

$$Q_c = \sum_{i=1}^n q_{C_i} \quad (16)$$

— **Crédibilité**

$$Q_{cr} = \frac{1}{n} \sum_{i=1}^n q_{cr_i} \quad (17)$$

où  $n$  est le nombre de services.

— **Fiabilité**

$$Q_F = \prod_{i=1}^n q_{F_i} \quad (18)$$

Cette étape est cruciale pour préparer les données en vue de l'évaluation de la fonction de fitness.

— **Évaluation de la population initiale**

Après avoir généré la population initiale, nous calculons la valeur de fitness de chaque composition en utilisant la fonction d'utilité suivante [11] :

$$F_{compi} = \sum_{j=1}^{nb\_qos} w_j \times Q'_j \quad (19)$$

où :

- $nb\_qos$  est le nombre de paramètres QoS;
- $w_j$  est le poids du  $j^{\text{ème}}$  attribut de QoS qui représente les préférences des utilisateurs;
- $Q'_j$  est la valeur agrégée du  $j^{\text{ème}}$  attribut de QoS.

cette fonction d'utilité ou bien également appeler fonction de fitness simplifie le choix de la meilleure composition afin de répondre aux exigences spécifiques de l'utilisateur [50]. Dans notre cas, nous cherchons une composition quasi-optimale qui minimise la fonction de fitness c'est-à-dire la composition ayant une valeur minimale sera choisi comme une meilleure composition .

### — Initialisation des compositions alpha, beta, delta

Après avoir calculé les valeurs de fitness pour chaque composition dans la population initiale, nous avons trié les compositions par ordre croissant de leurs valeurs de fitness. Ainsi, la composition avec la valeur de fitness minimale est sélectionnée comme la solution optimale, notée  $\alpha$ . La deuxième solution la plus adaptée est notée  $\beta$ , et la troisième est notée  $\delta$ .

### — Application des opérations de GWO

Tout d'abord, nous calculons les paramètres  $a$ ,  $\mathbf{A}$  et  $\mathbf{C}$  selon les équations suivantes :

$$a = 2 - \frac{2t}{T_{\max}} \quad (20)$$

$$A(t) = 2 \times a \times r_1 - a \quad (21)$$

$$C(t) = 2 \times r_2 \quad (22)$$

Ensuite, pour chaque composition de la population initiale, nous appliquons les opérations du GWO comme suit :

$$D_\alpha = |C_1 \cdot comp_\alpha - comp(t)| \quad (23)$$

$$D_\beta = |C_2 \cdot comp_\beta - comp(t)| \quad (24)$$

$$D_\delta = |C_3 \cdot comp_\delta - comp(t)| \quad (25)$$

$$comp_1 = comp_\alpha - A_1 \cdot D_\alpha \quad (26)$$

$$comp_2 = comp_\beta - A_2 \cdot D_\beta \quad (27)$$

$$comp_3 = comp_\delta - A_3 \cdot D_\delta \quad (28)$$

Enfin, nous mettons à jour la composition selon l'équation suivante :

$$comp(t+1) = \frac{comp_1 + comp_2 + comp_3}{3} \quad (29)$$

### — Mise à jour de la population

À l'issue du processus KM-GWO, l'algorithme sélectionne la composition ayant la plus petite valeur de fitness parmi celles générées au cours des itérations. À chaque itération, nous mettons à jour les trois meilleures solutions en comparant les valeurs de leur fitness avec celles des compositions précédentes, et  $\overrightarrow{comp}_\alpha$  est retenu comme la meilleure solution. L'algorithme GWO explore l'espace de recherche en produisant et en évaluant de nouvelles compositions à chaque itération. Progressivement, il converge vers une solution de plus en plus proche de l'optimum. Les étapes suivies par l'algorithme GWO sont détaillées dans l'algorithme 2.

## Chapitre 4 : Approche proposée : K-means with Grey Wolf Optimization Algorithm

---

### Algorithm 2: Algorithme de GWO

---

**Input:**  $n\_iter, Nbr\_cs, Max\_Nbrcs, Comp\_i, population\_size, poids$

**Output:** Meilleure composition de fitness

```
1 Début ;
2 for  $Nbr\_cs$  To  $Max\_Nbrcs$  do
3   Génération aléatoire de la population initiale ;
4   Évaluation de la population initiale en utilisant l'équation (19) ;
5   Initialisation des  $\overrightarrow{com\hat{p}}_\alpha, \overrightarrow{com\hat{p}}_\beta, \overrightarrow{com\hat{p}}_\delta$  ;
6   while  $t \leq n\_iter$  do
7     for chaque composition to  $population\_size$  do
8       Calcul des coefficients  $a, A$  et  $C$  selon les équations 20, 21, 22 ;
9       Application des opérations GWO selon les équations 23, 24, 25, 26, 27,
10      28 ;
11      Mise à jour de la composition en utilisant l'équation 29 ;
12    Fin pour;
13    Évaluation des nouvelles compositions avec l'équation (19) ;
14    Mise à jour des trois meilleures solutions ;
15     $t \leftarrow t + 1$ ;
16  end while ;
17  Retourner  $\overrightarrow{com\hat{p}}_\alpha$  et son fitness ;
18 Fin
```

---

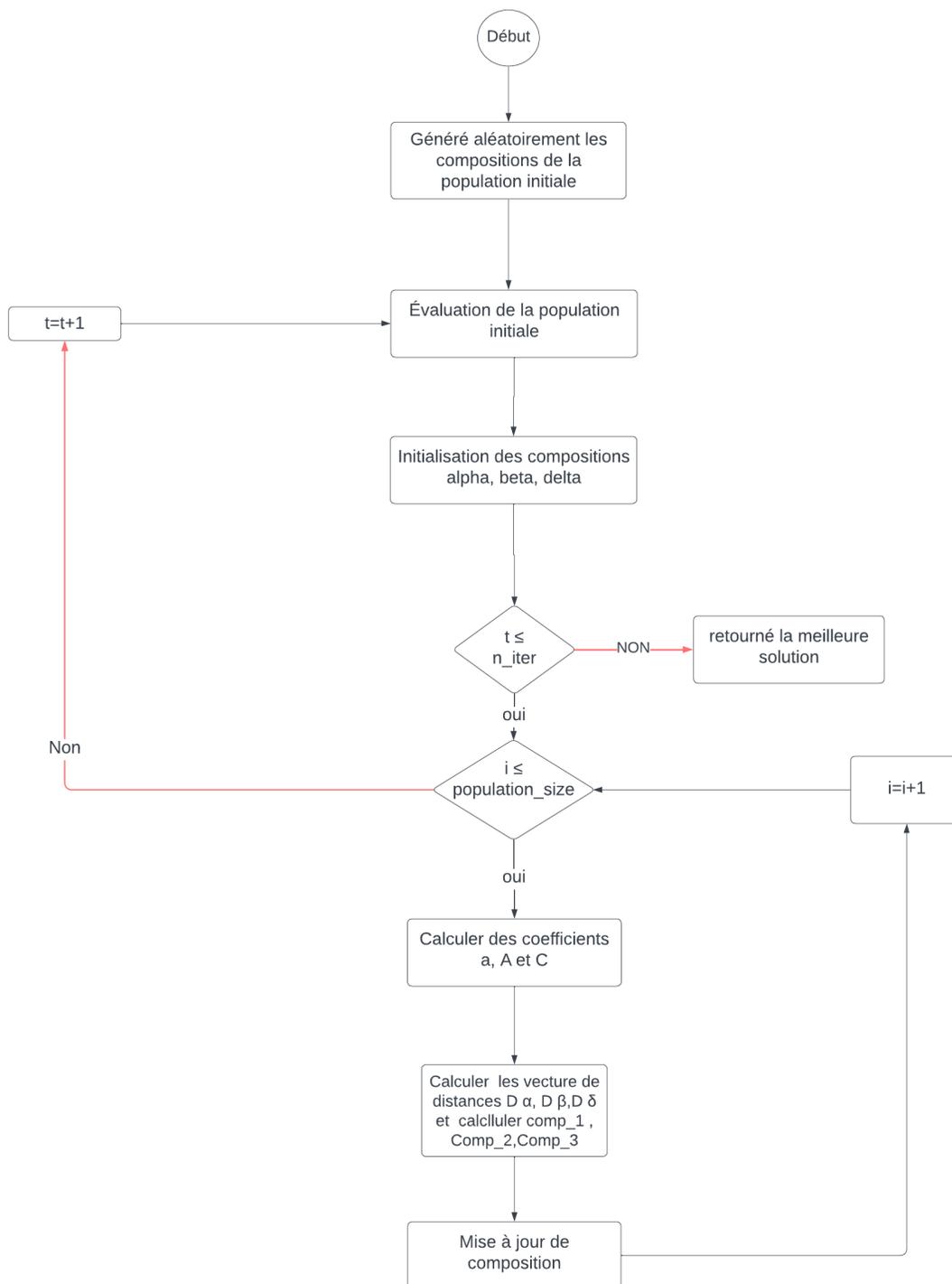


FIGURE 11 – Digramme de flux de l’algorithme GWO

### 4.6 Validation de l'approche

#### 4.6.1 Environnement et outils de développement

L'implémentation et l'évaluation de l'approche proposée ont été réalisées sur un ordinateur portable fonctionnant sous Windows 10 Professionnel. Cet ordinateur est équipé d'un processeur Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz, avec une fréquence turbo pouvant atteindre 2.11GHz, ainsi que d'un système d'exploitation 64 bits et un processeur x64.

##### **Anaconda3**

Anaconda3 est une distribution open-source des langages de programmation Python et R, principalement utilisée en science des données pour simplifier la gestion des packages et leur déploiement. Les versions des packages dans Anaconda sont gérées par le système de gestion de packages, Conda, qui analyse l'environnement actuel avant d'exécuter une installation pour éviter de perturber d'autres frameworks et packages [37].

Anaconda3 est un outil puissant pour les développeurs Python, car il simplifie la configuration, la gestion des packages et des environnements, tout en offrant un écosystème complet pour le développement en Python, en particulier dans les domaines de la science des données et du machine learning [37].

##### **JupyterLab**

JupyterLab est une application qui permet de créer, ouvrir et lire des scripts Python (ou notebooks) avec l'extension .ipynb. Inclus dans Anaconda3, il permet aux utilisateurs de découvrir et d'expérimenter ses fonctionnalités. C'est une version de nouvelle génération du Jupyter Notebook, offrant des fonctionnalités telles que Notebook, Editor, Terminal et Console [24].

##### **Visual Studio Code**

Visual Studio Code (VS Code) est un éditeur de code open-source développé par Microsoft, disponible pour Windows, macOS et Linux. Il prend en charge de nombreux lan-

gages de programmation tels que C, C++, Java, JavaScript, Python, etc.

### **Python**

Python est un langage de programmation simple et puissant, de plus en plus populaire, utilisé dans de nombreux domaines comme le développement de logiciels, l'analyse de données, le machine learning et le développement web. Il fonctionne sur différentes plateformes, telles que Windows, macOS et Linux, sans nécessiter de grandes modifications du code [51].

### **Bibliothèques**

#### **Pandas**

Pandas est une bibliothèque open-source pour le langage de programmation Python, principalement utilisée pour la manipulation et l'analyse de données. Elle fournit des structures de données flexibles et expressives, telles que les DataFrames et les Series, qui facilitent le traitement de données étiquetées et hétérogènes [37].

#### **NumPy**

NumPy (Numerical Python) est une bibliothèque fondamentale pour le calcul scientifique en Python. Elle fournit un support pour les tableaux multidimensionnels et des matrices, ainsi que des fonctions mathématiques de haut niveau pour effectuer des opérations sur ces tableaux [56].

#### **scikit-learn**

scikit-learn est une bibliothèque de machine learning pour Python. Elle comprend des algorithmes de classification, de régression et de clustering, ainsi que des outils pour la validation des modèles et le traitement des données [46].

### Matplotlib

Matplotlib est une bibliothèque de visualisation de données pour Python. Elle permet de produire des figures de qualité dans une variété de formats imprimables et interactifs [22].

#### 4.6.2 Description de Dataset

Le Dataset utilisé dans cette étude est nommé *IoTS<sub>D</sub>ataset* et il est téléchargeable depuis [33]. Les services IoT sont caractérisés par des attributs de qualité de service tels que le temps d'exécution, le coût du service, la réputation et la fiabilité. Chaque valeur d'attribut de qualité de service est générée de manière aléatoire par un algorithme spécifique dans les plages de valeurs définies pour chaque attribut. La taille totale du Dataset est de 8800 Ko.

Ce Dataset est organisé en plusieurs échelles de services IoT, chacune représentant une combinaison différente de tâches et de services candidats, notées *IoTS10X50*, *IoTS10X100*, *IoTS20X50*, *IoTS20X100*, *IoTS30X50* et *IoTS30X100*.

- *IoTS10X50* : Composé de 10 fichiers Excel représentant 10 tâches, où chaque tâche correspond à un service IoT abstrait, et chaque tâche contient 50 services IoT candidats. Ces candidats sont fonctionnellement similaires mais diffèrent en termes de qualité de service non fonctionnelle.
- *IoTS10X100* : Inclut également 10 fichiers Excel représentant 10 tâches, chaque tâche contenant 100 services IoT candidats.
- *IoTS20X50* : Composé de 20 fichiers Excel représentant 20 tâches, avec chaque tâche contenant 50 services IoT candidats.
- *IoTS20X100* : Similaire à *IoTS10X100* mais avec 20 tâches et 100 services IoT candidats chacune.
- *IoTS30X50* : Comprend 30 tâches contenant chacune 50 services IoT candidats.
- *IoTS30X100* : Similaire à *IoTS10X100* mais avec 30 tâches et 100 services IoT candidats chacune.

*IoTS<sub>D</sub>dataset* est composé de cinq (5) colonnes comme suit :

- Numéro IoT : indique le numéro de l'IoT (par exemple : 1).

## Chapitre 4 : Approche proposée : K-means with Grey Wolf Optimization Algorithm

---

- Temps d'exécution : représente le temps d'exécution (par exemple : 20).
- Coût du service : indique le coût du service (par exemple : 69).
- Crédibilité : représente la crédibilité (par exemple : 3).
- Fiabilité : indique la fiabilité (par exemple : 0,88).

La figure 12 présente un aperçu des données contenues dans le Dataset

IoTS Number	Execution Time	Service Cost	Credibility	Reliability
1	20	69	3	0.88
2	57	56	4	0.1
3	37	38	3	0.1
4	16	92	6	0.58
5	52	36	3	0.81
6	10	28	5	0.39
7	23	17	5	0.5
8	60	1	4	0.79
9	21	69	4	0.26
10	16	37	7	0.25
11	32	96	10	0.87
12	31	54	5	0.18
13	44	45	8	0.79
14	3	53	10	0.78
15	30	99	4	0.53
16	1	8	4	0.28
17	1	44	5	0.51
18	38	28	4	0.51
19	38	42	10	0.49
20	28	82	4	0.2

FIGURE 12 – IoT Dataset [33]

### 4.6.3 Prétraitement des données du *IoTS<sub>D</sub>ataset*

La préparation des données se fait en plusieurs étapes importantes :

- **Importation des données** : Avant la préparation des données, nous commençons par l'importation de notre ensemble de données, stocké dans des fichiers Excel. La figure 13 montre comment importer un ensemble de données à partir d'un fichier Excel.

```
# Fonction pour charger les données à partir de fichiers
def load_data(folder_path):
    all_data = []
    file_pattern = re.compile(r"T(\d+)_IoTS\.xlsx")

    # Créer une liste de tuples (nom_de_fichier, numéro)
    files_with_numbers = []
    for file_name in os.listdir(folder_path):
        match = file_pattern.match(file_name)
        if match:
            file_number = int(match.group(1))
            files_with_numbers.append((file_name, file_number))

    # Trier la liste de tuples par le numéro de fichier
    files_with_numbers.sort(key=lambda x: x[1])
```

FIGURE 13 – Import d'un Dataset

- **Prétraitement des données :** Les données importés dans l'étape précédentes ont été traités pour garantir que chaque service IoT soit identifié de manière unique et que toutes les colonnes redondantes ou vides soient supprimées. La figure 12 montre un exemple de la structure du dataset après cette préparation initiale.
- **Normalisation des données :** Nous avons pris soin de normaliser les données pour garantir une échelle uniforme des différents attributs. Cette étape est cruciale car elle permet aux algorithmes d'optimisation de fonctionner de manière optimale, même si les attributs ont des plages de valeurs initialement très différentes. En mettant à l'échelle les données, nous assurons que chaque attribut contribue de manière équitable au processus d'optimisation, ce qui améliore la performance et la précision des résultats.

## Chapitre 4 : Approche proposée : K-means with Grey Wolf Optimization Algorithm

---

```
Dataset sans normalisation :
  IoTS Number  Execution Time  Service Cost  Credibility  Reliability
0             1             20             69           3           0.88
1             2             57             56           4           0.10
2             3             37             38           3           0.10
3             4             16             92           6           0.58
4             5             52             36           3           0.81
..           ...           ...           ...           ...           ...
45            46             51             73           4           0.39
46            47             49              5           9           0.80
47            48              4             36           6           0.79
48            49              1             27           9           0.52
49            50             33             47           7           0.52
```

```
[2500 rows x 5 columns]
```

FIGURE 14 – Dataset avant la normalisation

```
  IoTS Number  Execution Time  Service Cost  Credibility  Reliability
0             1      0.322034      0.686869      0.000000      0.857143
1             2      0.949153      0.555556      0.142857      0.000000
2             3      0.610169      0.373737      0.000000      0.000000
3             4      0.254237      0.919192      0.428571      0.527473
4             5      0.864407      0.353535      0.000000      0.780220
..           ...           ...           ...           ...           ...
45            2496      0.847458      0.727273      0.142857      0.318681
46            2497      0.813559      0.040404      0.857143      0.769231
47            2498      0.050847      0.353535      0.428571      0.758242
48            2499      0.000000      0.262626      0.857143      0.461538
49            2500      0.542373      0.464646      0.571429      0.461538
```

```
[2500 rows x 5 columns]
```

FIGURE 15 – Dataset après la normalisation

### 4.6.4 Scénario de validation de KM-GWO

Pour tester l'efficacité de notre approche, nous avons décidé de l'appliquer au cas d'une sonnette vidéo connectée dans une maison intelligente. Cette sonnette permet aux utilisateurs de voir et de discuter avec leurs visiteurs à distance via leur smartphone, offrant ainsi une solution pratique pour garder un œil sur l'entrée de leur maison. Nous avons défini un plan de composition avec trois services abstraits, à savoir :

- **Service de transmission vidéo en direct** : Ce service permet de capturer et d'envoyer la vidéo en temps réel vers l'application mobile de l'utilisateur.
- **Service de notifications et alertes** : Ce service détecte des événements tels que l'appui sur la sonnette, ou des mouvements, et envoie des notifications à l'utilisateur.
- **Service d'archivage et d'analyse des données** : Ce service est responsable du stockage des vidéos et des journaux d'événements pour un accès futur.

La combinaison de ces trois services offre une solution de sécurité et de confort à l'utilisateur.



FIGURE 16 – plan de composition pour le scénario de KM-GWO

Chaque service abstrait du plan de composition, illustré à la figure 16, peut être implémenté par divers services IoT candidats qui partagent les mêmes fonctionnalités, mais avec des attributs QoS distincts. Le tableau 4 montre, pour chaque service abstrait dans le scénario, trois exemples de services IoT candidats. Cela représente un nombre de  $3^3$  combinaisons possibles de services IoT. L'approche KM-GWO proposée vise à identifier la meilleure combinaison de services qui satisfait aux exigences de qualité de service de l'utilisateur.

Service abstrait	Service candidat	Fournisseur
Service de transmission vidéo en direct	CS <sub>1</sub> <sup>1</sup>	Zoom Vidéo Communications
	CS <sub>1</sub> <sup>2</sup>	Microsoft Teams
	CS <sub>1</sub> <sup>3</sup>	Cisco Webex
Service de notifications et alertes	CS <sub>2</sub> <sup>1</sup>	Twilio
	CS <sub>2</sub> <sup>2</sup>	Pusher
	CS <sub>2</sub> <sup>3</sup>	OneSignal
Service d'archivage et d'analyse des données	CS <sub>3</sub> <sup>1</sup>	Amazon Web Services (AWS)
	CS <sub>3</sub> <sup>2</sup>	Google Cloud Platform - BigQuery
	CS <sub>3</sub> <sup>3</sup>	Microsoft Azure - Data Lake

TABLE 4 – Services candidats disponibles pour chaque service abstrait dans le scénario de composition

### 4.7 Application de KM-GWO

#### a) Application de K-Means pour le Clustering des Services IoT

Cette section décrit l'application de l'algorithme KMeans pour le regroupement des services IoT en clusters basés sur leurs attributs de qualité de service. Nous avons commencé par attribuer des poids spécifiques à chaque attribut QoS, définis comme suit :

$$W_i = [0.1, 0.2, 0.3, 0.4]$$

Ces poids correspondent respectivement au temps d'exécution, au coût du service, à la crédibilité et à la fiabilité, afin de refléter leur importance relative. Ensuite, une somme pondérée a été calculée pour chaque service, puis normalisée pour garantir une prise en compte équilibrée de chaque attribut dans le processus de clustering. Le nombre de clusters a été fixé à trois (3) pour cette analyse. Chaque service a ensuite été affecté à un cluster en fonction des résultats obtenus par l'algorithme KMeans.

#### b) Visualisation et évaluation des clusters

Après le clustering, nous avons visualisé les groupes obtenus sous forme de nuages de points. La figure ci-dessous illustre la classification initiale avec K-means :

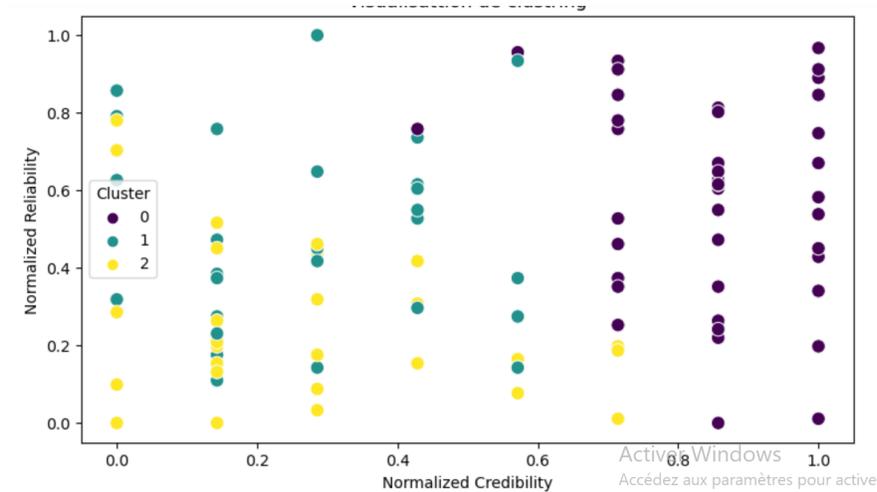


FIGURE 17 – Classification initiale avec K-means

Dans la classification initiale de K-means, on remarque que les points de chaque cluster ne sont pas bien séparés les uns des autres. Cela indique une mauvaise classification de l'algorithme K-means et, par conséquent, il est nécessaire de revoir les paramètres ou d'ajuster la méthode de clustering pour améliorer les résultats.

Pour améliorer la classification, nous avons introduit un nouvel attribut nommé QoS-somme. Celui-ci calcule la somme pondérée des colonnes QoS en multipliant chaque attribut par son poids et en les agréant. Cette amélioration a permis d'obtenir des résultats plus significatifs. La figure 18 montre la répartition des services IoT en fonction de leur QoS-somme, avec des clusters colorés différemment pour faciliter l'interprétation de la qualité du regroupement et la distribution des services au sein de chaque cluster.

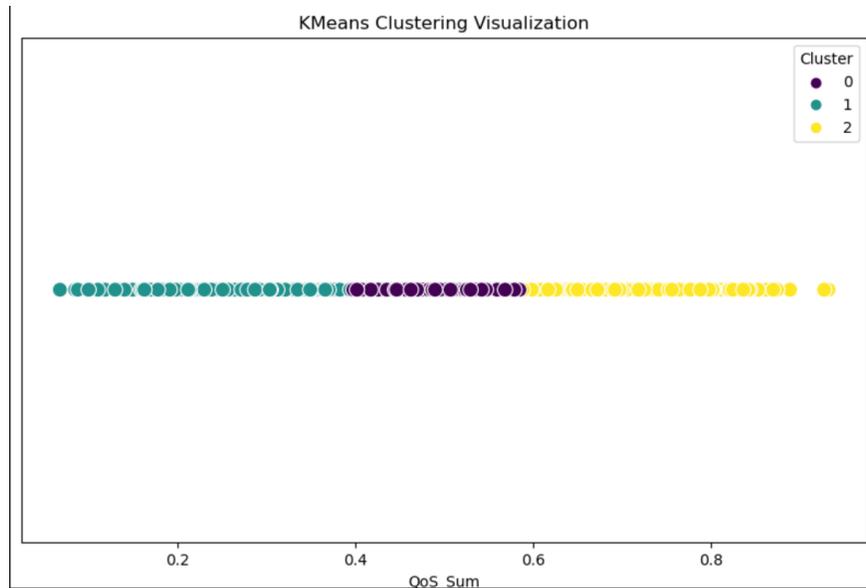


FIGURE 18 – Répartition des services IoT en fonction de leur QoS-somme

	IoTS Number	Execution Time	Service Cost	Credibility	Reliability	\
0	1	0.322034	0.686869	0.000000	0.857143	
1	2	0.949153	0.555556	0.142857	0.000000	
2	3	0.610169	0.373737	0.000000	0.000000	
3	4	0.254237	0.919192	0.428571	0.527473	
4	5	0.864407	0.353535	0.000000	0.780220	
..	...	...	...	...	...	
45	2496	0.847458	0.727273	0.142857	0.318681	
46	2497	0.813559	0.040404	0.857143	0.769231	
47	2498	0.050847	0.353535	0.428571	0.758242	
48	2499	0.000000	0.262626	0.857143	0.461538	
49	2500	0.542373	0.464646	0.571429	0.461538	

	QoS_Sum	Cluster
0	0.613325	2
1	0.370783	1
2	0.234155	1
3	0.580451	0
4	0.591030	2
..	...	...
45	0.529432	0
46	0.568240	0
47	0.462384	0
48	0.349118	1

FIGURE 19 – Répartition des Services dans les Clusters

Ces graphiques fournissent une vue d'ensemble de la distribution des services IoT en fonction de leur QoS-somme, avec chaque point coloré selon son cluster. Cette visualisation permet d'évaluer l'efficacité du regroupement réalisé par l'algorithme K-Means.

### c) Analyse des résultats du Clustering

Nous avons évalué la qualité des clusters formés par l'algorithme K-means en utilisant le score de silhouette. Ce score mesure la cohérence des clusters, en indiquant dans quelle mesure les services au sein d'un même cluster sont similaires entre eux par rapport à ceux des autres clusters. Le score varie de -1 à 1 : un score élevé, proche de 1, indique que les clusters sont bien séparés et distincts, tandis qu'un score proche de 0 signifie que les points se trouvent à la frontière entre deux clusters, tandis qu'un score négatif indique que les points sont mal classés, étant plus proches d'un autre cluster.

**Silhouette Score: 0.53**

FIGURE 20 – Score de Silhouette

Nous avons obtenu un score de silhouette de 0,53, ce qui signifie que les points au sein des clusters présentent une séparation modérée, indiquant une cohérence raisonnable entre les services d'un même cluster par rapport à ceux des autres clusters.

## 4.8 Application de GWO pour rechercher une composition proche de l'optimale

### 4.8.1 Initialisation de la Population

Nous avons créé la population initiale des compositions de services IoT en sélectionnant les services candidats regroupés en clusters. Chaque composition représente une combinaison potentielle de services, construite sur la base des attributs de qualité de service. Cette phase est essentielle, car elle établit les fondations sur lesquelles nous allons appliquer nos processus d'optimisation. En générant une population variée, nous nous assurons d'explorer efficacement l'espace de recherche, ce qui permet aux algorithmes d'optimisation de travailler avec une large gamme de solutions possibles.



```

Individual 7:
{'IoT Number': 661.0, 'Execution Time': 0.01694915254237288, 'Service Cost': 0.0303030303030304, 'Credibility':
0.7142857142857142, 'Reliability': 0.802197802197802, 'QoS_Sum': 0.5415849687036127, 'Cluster': 0.0}
{'IoT Number': 293.0, 'Execution Time': 0.1694915254237288, 'Service Cost': 0.010101010101010102, 'Credibility':
0.9999999999999998, 'Reliability': 0.9780219780219779, 'QoS_Sum': 0.7261171973036379, 'Cluster': 1.0}
{'IoT Number': 1655.0, 'Execution Time': 0.03389830508474577, 'Service Cost': 0.0202020202020202, 'Credibility':
0.5714285714285714, 'Reliability': 0.3956043956043956, 'QoS_Sum': 0.3384701927074808, 'Cluster': 2.0}

Individual 8:
{'IoT Number': 1938.0, 'Execution Time': 0.03389830508474577, 'Service Cost': 0.0202020202020202, 'Credibility':
0.857142857142857, 'Reliability': 0.7802197802197801, 'QoS_Sum': 0.5780306322679203, 'Cluster': 0.0}
{'IoT Number': 1531.0, 'Execution Time': 0.01694915254237288, 'Service Cost': 0.010101010101010102, 'Credibility':
0.9999999999999998, 'Reliability': 0.9780219780219779, 'QoS_Sum': 0.6956087227273666, 'Cluster': 1.0}
{'IoT Number': 1655.0, 'Execution Time': 0.03389830508474577, 'Service Cost': 0.0202020202020202, 'Credibility':
0.5714285714285714, 'Reliability': 0.3956043956043956, 'QoS_Sum': 0.3384701927074808, 'Cluster': 2.0}

Individual 9:
{'IoT Number': 1938.0, 'Execution Time': 0.03389830508474577, 'Service Cost': 0.0202020202020202, 'Credibility':
0.857142857142857, 'Reliability': 0.7802197802197801, 'QoS_Sum': 0.5780306322679203, 'Cluster': 0.0}
{'IoT Number': 1746.0, 'Execution Time': 0.22033898305084745, 'Service Cost': 0.25252525252525254, 'Credibility':
0.9999999999999998, 'Reliability': 0.9999999999999998, 'QoS_Sum': 0.7693203218626946, 'Cluster': 1.0}
{'IoT Number': 1655.0, 'Execution Time': 0.03389830508474577, 'Service Cost': 0.0202020202020202, 'Credibility':
0.5714285714285714, 'Reliability': 0.3956043956043956, 'QoS_Sum': 0.3384701927074808, 'Cluster': 2.0}

```

FIGURE 23 – Initialisation de la population3

### — Calcul de $V_{QoS}$ de chaque composition de la population

Après avoir initialisé la population, nous avons combiné les valeurs individuelles de QoS pour chaque attribut afin d’analyser les compositions de services IoT. Cette étape consiste à calculer des mesures globales pour chaque composition, en tenant compte des différents attributs de qualité de service. L’agrégation permet de condenser les informations sur les services IoT en une vue d’ensemble, facilitant ainsi la comparaison et l’évaluation des diverses compositions. Cette étape est cruciale pour préparer les données en vue de l’évaluation de la fonction de fitness.

	Execution Time	Service Cost	Credibility	Reliability
0	0.101695	0.232323	0.809524	0.280616
1	0.271186	0.343434	0.761905	0.369521
2	0.305085	0.262626	0.761905	0.357339
3	0.338983	0.393939	0.809524	0.277463
4	0.067797	0.101010	0.761905	0.361399
5	0.271186	0.343434	0.761905	0.369521
6	0.220339	0.060606	0.761905	0.310378
7	0.084746	0.050505	0.809524	0.301875
8	0.288136	0.292929	0.809524	0.308658
9	0.271186	0.131313	0.809524	0.281524
10	0.254237	0.181818	0.761905	0.337035
11	0.084746	0.050505	0.809524	0.301875
12	0.288136	0.131313	0.761905	0.253768
13	0.271186	0.131313	0.809524	0.281524
14	0.288136	0.292929	0.809524	0.308658
15	0.067797	0.060606	0.761905	0.310378
16	0.288136	0.313131	0.809524	0.261698
17	0.288136	0.313131	0.809524	0.261698
18	0.288136	0.131313	0.761905	0.253768
19	0.288136	0.313131	0.809524	0.261698

FIGURE 24 – Résultat d’agrégation des valeurs de QoS

### — Normalisation des Données après Agrégation

Avant l’évaluation de la fonction de fitness, nous avons normalisé les données agrégées pour garantir une échelle uniforme des attributs agrégés. Cette normalisation

est essentielle pour assurer que chaque attribut contribue de manière équitable à l'évaluation des compositions de services.

	Execution Time	Service Cost	Credibility	Reliability
0	0.600000	0.000000	0.0	1.000000
1	0.866667	1.000000	0.0	0.996709
2	0.933333	0.903226	1.0	0.068285
3	0.933333	0.838710	1.0	0.472645
4	0.800000	0.483871	0.0	0.716989
5	1.000000	0.838710	0.0	0.210613
6	0.933333	0.838710	1.0	0.472645
7	0.000000	0.000000	0.0	1.000000
8	0.066667	0.096774	0.0	0.487454
9	1.000000	0.741935	0.0	0.891814
10	0.866667	1.000000	0.0	0.996709
11	0.200000	0.064516	0.0	0.157960
12	0.933333	0.903226	1.0	0.068285
13	0.933333	0.838710	1.0	0.472645
14	0.600000	0.000000	0.0	1.000000
15	0.066667	0.225806	0.0	0.926779
16	0.933333	0.322581	0.0	0.000000
17	0.866667	0.322581	1.0	0.238996
18	0.066667	0.096774	0.0	0.487454
19	0.933333	0.516129	0.0	0.964212

FIGURE 25 – Normalisation après l'agrégation

## 4.8.2 Évaluation des composition (Fitness)

L'évaluation des compositions à l'aide de la fonction de fitness consiste à mesurer la qualité des différentes compositions de services IoT . Cette fonction permet d'analyser chaque composition en tenant compte de divers attributs de qualité de service, tels que le temps d'exécution, le coût, la crédibilité et la fiabilité. Les compositions qui ont une valeur minimale sont considérées comme les plus performantes, nous aidant ainsi à sélectionner les solutions optimales parmi les options disponibles.

Rappelons que nous calculons les valeurs de fitness selon la formule suivante :

$$F_{\text{compi}} = \sum_{j=1}^{\text{nb\_qos}} w_j \times Q'_j \quad (30)$$

	Execution Time	Service Cost	Credibility	Reliability	Fitness
0	0.600000	0.000000	0.0	1.000000	0.520000
1	0.866667	1.000000	0.0	0.996709	0.672017
2	0.933333	0.903226	1.0	0.068285	0.604303
3	0.933333	0.838710	1.0	0.472645	0.759596
4	0.800000	0.483871	0.0	0.716989	0.495183
5	1.000000	0.838710	0.0	0.210613	0.368116
6	0.933333	0.838710	1.0	0.472645	0.759596
7	0.000000	0.000000	0.0	1.000000	0.400000
8	0.066667	0.096774	0.0	0.487454	0.217992
9	1.000000	0.741935	0.0	0.891814	0.630919
10	0.866667	1.000000	0.0	0.996709	0.672017
11	0.200000	0.064516	0.0	0.157960	0.109635
12	0.933333	0.903226	1.0	0.068285	0.604303
13	0.933333	0.838710	1.0	0.472645	0.759596
14	0.600000	0.000000	0.0	1.000000	0.520000
15	0.066667	0.225806	0.0	0.926779	0.406626
16	0.933333	0.322581	0.0	0.000000	0.218925
17	0.866667	0.322581	1.0	0.238996	0.601190
18	0.066667	0.096774	0.0	0.487454	0.217992
19	0.933333	0.516129	0.0	0.964212	0.623964

FIGURE 26 – Evaluation des compositions avec fitness

## 4.9 Recherche d’une composition quasi-optimal via GWO

Nous abordons à présent la seconde phase de notre approche, qui consiste à utiliser l’algorithme Grey Wolf Optimizer pour identifier la meilleure composition. Les résultats de cette application sont présentés ci-dessus.

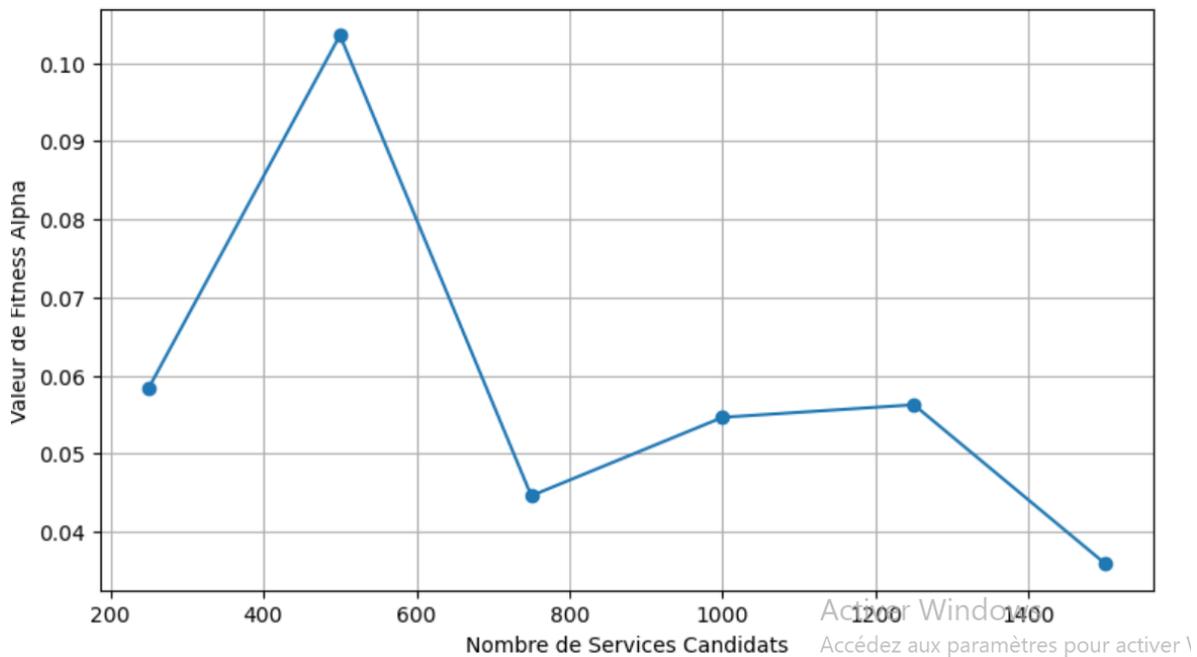


FIGURE 27 – Graphe de fitness en fonction de services candidats

### Phase d’observation des résultats du graphe

On observe que, dans le graphe de la fitness en fonction des services candidats, au début de l’algorithme GWO avec 250 services candidats, il renvoie le meilleur score de

## Chapitre 4 : Approche proposée : K-means with Grey Wolf Optimization Algorithm

---

fitness. Cependant, à mesure que le nombre de services candidats augmente, ce score fluctue avant de chuter brusquement. Ces variations des valeurs de fitness sont dues aux opérateurs du GWO.

```
-----  
best_Solution:  
0 \  
0 {'IoTS Number': 1375.0, 'Execution Time': 0.01...  
1 \  
0 {'IoTS Number': 24.0, 'Execution Time': 0.2542...  
2  
0 {'IoTS Number': 1655.0, 'Execution Time': 0.03...  
best_score: 0.02
```

FIGURE 28 – Meilleure solution

Pour démontrer l'efficacité de notre approche, nous l'avons comparée au GWO. Le tableau 5 présente un récapitulatif des valeurs de fitness obtenues par KM-GWO en comparaison avec GWO.

Nbrcs	250	500	750	1000	1250	1500
KM-GWO	0,08	0,05	0,06	0,07	0,02	0,04
GWO	0,38	0,45	0,46	0,38	0,43	0,35

TABLE 5 – Évaluation de Valeurs de fitness de KM-GWO et de GWO

Le graphique illustré à la figure 29 montre que KM-GWO offre une meilleure composition que GWO en termes de fitness. Cela confirme que la combinaison du GWO avec K-means améliore les performances du GWO.

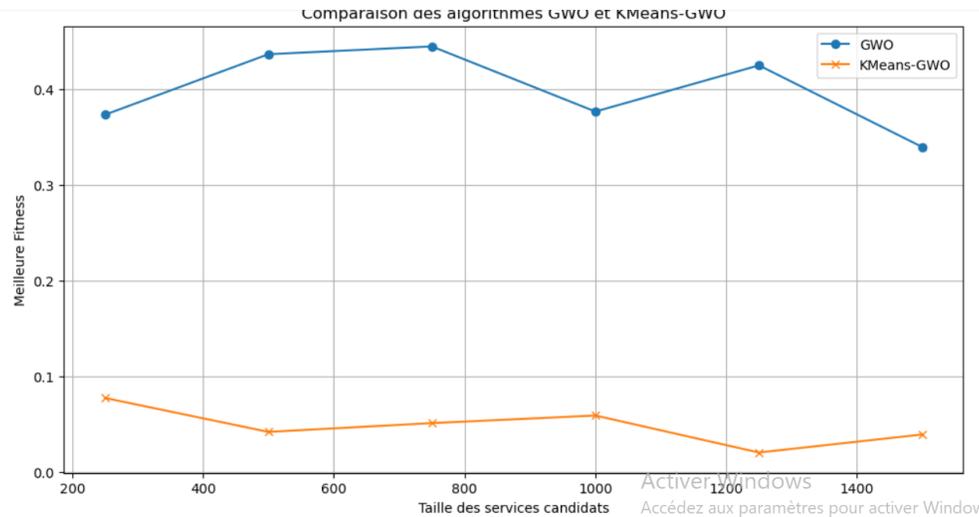


FIGURE 29 – Évaluation des valeurs de fitness des KM-GWO et de GWO traditionnel.

### 4.10 Conclusion

Dans ce chapitre, nous avons présenté l'approche hybride KM-GWO pour la composition des services IoT avec prise en compte des préférences QoS de l'utilisateur. Pour cela nous avons choisis de combiner la technique de clustering K-Means avec l'algorithme Grey Wolf Optimization. Nous avons d'abord étudié les principes du K-Means, utilisé pour classer les services en différents groupes en fonction de leurs attributs de qualité de service. Cette étape a été suivie par l'application de l'algorithme GWO, qui a optimisé les compositions de services en affinant les solutions grâce à des mécanismes de recherche inspirés de la chasse des loups gris.

### Conclusion Générale

Dans ce mémoire, nous avons exploré le domaine de l'Internet des Objets et son rôle essentiel dans l'évolution technologique ainsi que son impact sur la société contemporaine. L'IoT permet l'interconnexion de divers dispositifs via Internet, facilitant une interaction fluide entre le monde physique et le monde numérique. Notre recherche s'est concentrée sur la composition de services IoT, un processus essentiel pour répondre aux demandes complexes des utilisateurs en combinant plusieurs services individuels. Nous avons d'abord établi un cadre théorique en introduisant les concepts fondamentaux liés au problème de composition de services IoT, puis examiné diverses approches existantes pour la composition de services IoT, en prenant en compte les critères de Qualité de Service.

Dans ce travail, nous avons proposé une approche combinant l'algorithme K-means avec l'algorithme Grey Wolf Optimizer, que nous avons nommée KM-GWO. Cette approche vise à résoudre le problème de sélection de composition de services en tenant compte des QoS. L'algorithme K-means permet de classer les services candidats ayant les meilleures valeurs de QoS en trois clusters : jaune, vert et violet. Cette classification nous a permis de réduire l'espace de recherche. Cette réduction a optimisé le temps d'exécution de KM-GWO tout en maintenant une bonne qualité de la solution retournée. Pour valider notre approche, nous avons choisi un scénario de sonnette vidéo dans une maison intelligente. Les résultats d'évaluation de KM-GWO, comparés à ceux de l'algorithme GWO original sans l'utilisation de K-means, montrent que notre approche propose de meilleurs résultats en termes de valeurs de fitness.

### Perspectives

Dans le cadre de l'évolution de notre travail, plusieurs perspectives intéressantes peuvent être envisagées :

- Création d'un dataset réel pour les services IoT : Développer un dataset détaillé avec des données réelles pour améliorer la recherche et tester les approches proposées de manière plus précise.
- Exploration de méthodes hybrides : Combiner des méta-heuristiques avancées (comme les algorithmes génétiques) avec des techniques d'apprentissage profond pour améliorer la classification des services IoT.
- Implémentation dans des environnements réels : Tester l'approche KM-GWO dans des scénarios réels pour valider son efficacité pratique et identifier des améliorations potentielles.

## Références

- [1] Ines ALAYA, Christine SOLNON et Khaled GHEDIRA. « Optimisation par colonies de fourmis pour le problème du sac-à-dos multi-dimensionnel ». In : *Revue des Sciences et Technologies de l'Information - Série TSI: Technique et Science Informatiques* 3–4 (2007). HAL Id: hal-01542507, Submitted on 25 Mar 2021, p. 371-390. URL : <https://hal.science/hal-01542507>.
- [2] A. ALI MOHAMED. « Les avantages de l'Internet des objets pour les entreprises ». In : *LeMagIT* (2023). Consulté le 10 mai 2024. URL : <https://www.lemagit.fr/conseil/Les-avantages-et-les-inconvenients-de-lIoT-pour-les-entreprises>.
- [3] Mohammad ALRIFAI et Thomas RISSE. « Combining global optimization with local selection for efficient QoS-aware service composition ». In : *Proceedings of the 18th International World Wide Web Conference (WWW'09)* (2009), p. 881-890. DOI : 10.1145/1526709.1526828. URL : <https://doi.org/10.1145/1526709.1526828>.
- [4] Idir AOUDIA. « Composition adaptative de services pour l'Internet des objets ». Thèse de doctorat. Université de Mohamed Kheider Biskra, 2022.
- [5] Béa ARRUABARRENA. « Objets connectés : penser les enjeux des technologies connectées sous l'angle de la médiation infocommunicationnelle ». In : *Tic& société* 15.2-3 (2022), p. 9-35. DOI : 10.4000/ticetsociete.6262. URL : <https://journals.openedition.org/ticetsociete/6262?lang=en>.
- [6] KyeongDeok BAEK et In-Young KO. « Dynamic and Effect-Driven Output Service Selection for IoT Environments Using Deep Reinforcement Learning ». In : *IEEE Internet of Things Journal* 10 (jan. 2022), p. 3339-3355. DOI : 10.1109/JIOT.2022.3211270. URL : <https://ieeexplore.ieee.org/document/9908162>.
- [7] Eddy BAJIC. « Localisation et identification de ressources industrielles par l'Internet des objets ». In : *Journal of Industrial IoT* (avr. 2018), p. 19-27. DOI : 10.1016/j.compind.2018.04.015. URL : <https://hal.science/hal-02111606>.
- [8] Gaurav BARANWAL, Manisha SINGH et Deo Prakash VIDYARTHI. « A framework for IoT service selection ». In : *The Journal of Supercomputing* (nov. 2019). DOI : 10.1007/s11227-019-03076-1. URL : <https://link.springer.com/article/10.1007/s11227-019-03076-1>.
- [9] BENREBAH. « Gateway IoT de pilotage et de surveillance des capteurs domestiques via le protocole MQTT ». In : *Nom du Journal ou de la Conférence* (2023). Consulté le 10 mai 2024.

- [10] A. BOUALOUACHE. « La sécurité de l'Internet des Objets (IoT) ». In : *Mémoire de Master, Université de Tiaret* (2022). URL : <http://dspace.univ-tiaret.dz/bitstream/123456789/2580/1/TH.M.GE.FR.2022.16.pdf>.
- [11] Rabah BOUCETTI, Sofiane Mounine HEMAM et Ouassila HIOUAL. « An approach based on genetic algorithms and neural networks for QoS-aware IoT services composition ». In : *Journal of King Saud University Computer and Information Sciences* 34.8 (sept. 2022), p. 5619-5632. DOI : 10.1016/j.jksuci.2022.02.012. URL : <https://doi.org/10.1016/j.jksuci.2022.02.012>.
- [12] Zheng-Yi CHAI, Meng-Meng DU et Guo-Zhi SONG. « A fast energy-centered and QoS-aware service composition approach for Internet of Things ». In : *Applied Soft Computing* 100 (mars 2021), p. 106914. DOI : 10.1016/j.asoc.2020.106914. URL : <https://doi.org/10.1016/j.asoc.2020.106914>.
- [13] Ke CHEN et al. « Cuckoo Coupled Improved Grey Wolf Algorithm for PID Parameter Tuning ». In : *Applied Sciences* 13.23 (déc. 2023). Published: 4 December 2023, p. 12944. DOI : 10.3390/app132312944. URL : <https://doi.org/10.3390/app132312944>.
- [14] J. CRACKNELL. « Comment la technologie IoT favorise le développement durable ». In : *Tele2 IoT* (2023). URL : <https://tele2iot.com/fr/blog/how-iot-technology-enables-sustainability/>.
- [15] M. D'ANGELO et al. « Decentralized learning for self-adaptive QoS-aware service assembly ». In : *Future Generation Computer Systems* 108 (2020), p. 210-227.
- [16] J. DELARBRE. « Interopérabilité dans l'Internet des Objets : défis et solutions ». In : *Trialog* (2022). URL : <https://www.trialog.com/fr/interoperabilite-semantique-basee-sur-linternet-des-objets-robotiques-application-aux-soins-de-sante-intelligents-et-au-bien-etre-emotionnel/>.
- [17] A. FUTUR. « L'Internet des Objets : Une croissance explosive des appareils connectés ». In : *Technologies et Innovations* (2024). Consulté le 10 mai 2024. URL : <https://www.techinnovations.com/iot-croissance-explosive>.
- [18] GEORGES. *Anatomie d'un device IoT : tout ce qu'il faut savoir*. <https://www.matooma.com/fr/s-informer/actualites-iot-m2m/device-iot>. Consulté le 20 février 2024.
- [19] Yazid GHERBAL. « Une approche de composition des services IoT pour l'agriculture ». Mémoire de Master. Université de Biskra, 2021. URL : [http://archives.univ-biskra.dz/bitstream/123456789/21315/1/Yazid\\_Gherbal.pdf](http://archives.univ-biskra.dz/bitstream/123456789/21315/1/Yazid_Gherbal.pdf).

- [20] Malika HANICHE et Nabila TABRAIT. *Internet des objets dans le domaine de l'agriculture de demain*. URL consultée le 21/01/2024. 2019. URL : <https://dspace.ummo.dz/server/api/core/bitstreams/9788d881-ebdb-4aec-99a7-3d322f73b16e/content>.
- [21] C. HATTINGH et M. MATTHEE. « IoT Frameworks and Complexity ». In : *ResearchGate* (2023). URL : [https://www.researchgate.net/publication/335168187\\_IoT\\_Frameworks\\_and\\_Complexity](https://www.researchgate.net/publication/335168187_IoT_Frameworks_and_Complexity).
- [22] J. D. HUNTER. « Matplotlib: A 2D Graphics Environment ». In : *Computing in Science & Engineering* 9.3 (2007), p. 90-95. DOI : 10.1109/MCSE.2007.55.
- [23] Alan JEFFARES. *Supervised vs Unsupervised Learning in 3 Minutes*. Published in Towards Data Science, 3 min read. Juill. 2018. URL : <https://towardsdatascience.com/supervised-vs-unsupervised-learning-in-3-minutes-34dbd2e02a34> (visité le 27/07/2024).
- [24] Project JUPYTER. *JupyterLab*. Consulté le 10 mai 2024. 2023. URL : <https://jupyterlab.readthedocs.io/en/latest/>.
- [25] Meysam KARIMI et Seyed Morteza BABAMIR. « QoS-aware web service composition using Gray Wolf Optimizer ». In : *International Journal of Information and Communication Technology Research* 9.1 (mars 2017), p. 9-16. URL : <http://ijict.itrc.ac.ir/article-1-44-en.html>.
- [26] N. KASHYAP, A. C. KUMARI et R. CHHIKARA. « Multi-objective Optimization using NSGA II for service composition in IoT ». In : *Procedia Computer Science* 167 (2020), p. 1928-1933. DOI : 10.1016/j.procs.2020.03.214.
- [27] Karima KHADIR et al. « A Genetic Algorithm-Based Approach for Fluctuating QoS Aware Selection of IoT Services ». In : *IEEE Access* 10.5 (jan. 2022). Accessed: February 17, 2022, p. 1-1. DOI : 10.1109/ACCESS.2022.3145853. URL : [https://www.researchgate.net/publication/358420149\\_A\\_Genetic\\_Algorithm-Based\\_Approach\\_for\\_Fluctuating\\_QoS\\_Aware\\_Selection\\_of\\_IoT\\_Services](https://www.researchgate.net/publication/358420149_A_Genetic_Algorithm-Based_Approach_for_Fluctuating_QoS_Aware_Selection_of_IoT_Services).
- [28] M. E. KHANOUCHE et al. « Energy-centered and QoS-aware services selection for Internet of Things ». In : *IEEE Transactions on Automation Science and Engineering* 13.4 (2016), p. 1256-1269. DOI : 10.1109/TASE.2016.2539240.
- [29] Mohamed Essaid KHANOUCHE et al. « Clustering-based and QoS-aware services composition algorithm for ambient intelligence ». In : *Information Sciences* 482 (mai 2019), p. 419-439. DOI : 10.1016/j.ins.2019.01.015. URL : <https://www.sciencedirect.com/science/article/abs/pii/S0020025519300155?via%3Dihub>.

- [30] Eugenia KOVATCHEVA. « User Experience Design Models for Internet of Things ». In : *Serdica Journal of Computing* 12.1-2 (déc. 2018), p. 65-82. DOI : 10.55630/sjc.2018.12.65-82. URL : <https://doi.org/10.55630/sjc.2018.12.65-82>.
- [31] S. LE DIGABEL. *Introduction aux métaheuristiques*. MTH6311, École Polytechnique de Montréal, H2018 (v2). 2018. URL : [https://www.gerad.ca/Sebastien.Le.Digabel/MTH6311/5\\_Introduction\\_Metaheuristiques.pdf](https://www.gerad.ca/Sebastien.Le.Digabel/MTH6311/5_Introduction_Metaheuristiques.pdf).
- [32] Khai Loon LEE et al. « Investigating the impact of benefits and challenges of IOT adoption on supply chain performance and organizational performance: An empirical study in Malaysia ». In : *Uncertain Supply Chain Management* 10.2 (2022), p. 537-550. DOI : 10.5267/j.uscm.2021.11.009. URL : <https://doi.org/10.5267/j.uscm.2021.11.009>.
- [33] Si LI. *IoTS-Dataset: QoS data about IoT services*. Déc. 2023. DOI : 10.5281/zenodo.10440967. URL : <https://zenodo.org/records/10440967>.
- [34] Amel LYLA. « Prévisions du marché mondial de l’IoT à l’horizon 2030 ». In : *Rapport d’étude de marché* (2023). Consulté le 10 mai 2024. URL : <https://www.rapportdetude.com/previsions-marche-iot-2030>.
- [35] H. B. MAHFOUDH et al. « Coordination model with reinforcement learning for ensuring reliable on-demand services in collective adaptive systems ». In : *Proceedings of the International Symposium on Leveraging Applications of Formal Methods*. Springer, 2018, p. 257-273.
- [36] Yassine MAMOU. « L’Edge Computing : la nouvelle frontière de l’IoT ». In : *Revue Technique et Scientifique* 42.3 (2024), p. 12-18. URL : <https://www.revuetechmique.com/article/edge-computing-iot-frontiere>.
- [37] Wes MCKINNEY. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O’Reilly Media, 2012.
- [38] Seyedali MIRJALILI, Seyed Mohammad MIRJALILI et Andrew LEWIS. « Grey Wolf Optimizer ». In : *Advances in Engineering Software* 69 (mars 2014), p. 46-61. DOI : 10.1016/j.advengsoft.2013.12.007. URL : <https://www.sciencedirect.com/science/article/abs/pii/S0965997813001853?via%3Dihub>.
- [39] R. MOUHA. *Internet of Things (IoT)*. <https://www.scirp.org/journal/paperinformation?paperid=108574>. Consulté le 30 mars. 2024.
- [40] Radouan Ait Radouan Ait MOUHA. « Internet of Things (IoT) ». In : *Journal of Data Analysis and Information Processing* 09.02 (jan. 2021), p. 77-101. DOI : 10.4236/jdaip.2021.92006. URL : [https://www.researchgate.net/publication/351003790\\_Internet\\_of\\_Things\\_IoT](https://www.researchgate.net/publication/351003790_Internet_of_Things_IoT).

- [41] Mohammad H. NADIMI-SHAHRAKI et al. « A systematic review of applying grey wolf optimizer, its variants, and its developments in different Internet of Things applications ». In : *Internet of Things* 26 (juill. 2024), p. 101135. DOI : 10.1016/j.iot.2024.101135. URL : <https://www.sciencedirect.com/science/article/abs/pii/S2542660524000775?via%3Dihub>.
- [42] J. NOYER. « L'Internet des Objets, l'Internet of "Everything" : quelques remarques sur l'intensification du plissement numérique du monde ». In : *Internet des Objets* 17.1 (2017). URL : <https://doi.org/10.21494/iste.op.2017.0134>.
- [43] I. H. OSMAN et G. LAPORTE. « Metaheuristics: a bibliography ». In : *Annals of Operations Research* 63 (1996), p. 513-623.
- [44] M. OUZZANI et A. BOUGUETTAYA. « Efficient access to web services ». In : *IEEE Internet Computing* 11.2 (2007), p. 34-44.
- [45] B. OVHCLOUD. *Qu'est-ce que l'Internet des objets ?* <https://www.ovhcloud.com/fr/about-us/>. Consulté le 2 février. 2024.
- [46] F. PEDREGOSA et al. « Scikit-learn: Machine Learning in Python ». In : *Journal of Machine Learning Research* 12 (2011), p. 2825-2830. URL : <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>.
- [47] Aleksandr PETRUSEV et al. « L'intelligence artificielle pour la gestion des réseaux de distribution ». In : *La Revue de l'Électricité et de l'Électronique* (2022). HAL Id: hal-03720550. URL : [https://www.researchgate.net/publication/362252020\\_L%27intelligence\\_artificielle\\_pour\\_la\\_gestion\\_des\\_reseaux\\_de\\_distribution](https://www.researchgate.net/publication/362252020_L%27intelligence_artificielle_pour_la_gestion_des_reseaux_de_distribution).
- [48] Yihui QIU, Xiaoxiao YANG et Shuixuan CHEN. « An improved gray wolf optimization algorithm solving functional optimization and engineering design problems ». In : *Scientific Reports* 14.1 (juin 2024). DOI : 10.1038/s41598-024-64526-2. URL : <https://doi.org/10.1038/s41598-024-64526-2>.
- [49] Meroua RIZI, Soumia MAAZAOUI et Soumia ZERTAL. « Une nouvelle approche basée deep learning pour la classification des services cloud ». In : *Revue non spécifiée* (2021). Journal non précisé.
- [50] Samia SADOUKI NÉE CHIBANI. « Métaheuristiques bio-inspirées pour la sélection basée sur la QoS dans une composition de service web ». Thèse de doctorat. Thèse de doct. Université A. MIRA BEJAIA, 2018.
- [51] Futura SCIENCES. *Définition | Python*. Consulté le 5 mars 2024. 2024. URL : <https://www.futura-sciences.com/tech/definitions/informatique-python-19349>.

- [52] Satyajit SINHA. *State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally*. Consulté le 11 août 2024. Mai 2023. URL : <https://iot-analytics.com/number-connected-iot-devices/>.
- [53] Faker SKANDRANI. *Architecture IoT : L'essentiel à savoir*. Oct. 2023. URL : <https://iotindustriel.com/iot-iiot/architecture-iot-lessentiel-a-savoir/>.
- [54] El-Ghazali TALBI. « Metaheuristics: From Design to Implementation ». In : *Wiley* (2009). Ce livre fournit une étude complète des méta-heuristiques, comparant leurs avantages avec les approches déterministes., p. 1-624. URL : <https://doi.org/10.1002/9780470496916>.
- [55] Zhengyi TANG et al. « IoT service composition based on improved Shuffled Frog Leaping Algorithm ». In : *Heliyon* 10.7 (2024), e28087. DOI : 10.1016/j.heliyon.2024.e28087. URL : <https://doi.org/10.1016/j.heliyon.2024.e28087>.
- [56] S. VAN DER WALT, S. C. COLBERT et G. VAROQUAUX. « The NumPy Array: A Structure for Efficient Numerical Computation ». In : *Computing in Science & Engineering* 13.2 (2011), p. 22-30. DOI : 10.1109/MCSE.2011.37.
- [57] Hongbing WANG, Shunshun PENG et Qi YU. « A parallel refined probabilistic approach for QoS-aware service composition ». In : *Future Generation Computer Systems* 98 (sept. 2019), p. 609-626. DOI : 10.1016/j.future.2019.03.053. URL : <https://doi.org/10.1016/j.future.2019.03.053>.
- [58] Yuanyuan XIAO et al. « Large-scale web service composition based on improved grey wolf optimizer algorithm ». In : *Journal of Computer Applications* 42.10 (2022), p. 3162.
- [59] Jianqi YU. « Ligne de produits dynamique pour les applications à services ». Thèse de doct. Université Joseph-Fourier - Grenoble I, 2010. URL : <https://tel.archives-ouvertes.fr/tel-00493355>.
- [60] Ahmed ZEBOUCHI et Youcef AKLOUF. « pRTMNSGA-III: a novel multi-objective algorithm for QoS-aware multi-cloud IoT service selection ». In : *Annals of Telecommunications/Annales des Télécommunications* (jan. 2024). DOI : 10.1007/s12243-023-01006-0. URL : <https://doi.org/10.1007/s12243-023-01006-0>.
- [61] Zhao ZHONG et Christine BALAGUÉ. « Comprendre les objets connectés grand public : proposition d'une taxonomie centrée sur l'utilisateur ». In : *Vie & Sciences de l'Entreprise* N° 211-212.1 (juill. 2021), p. 70-90. DOI : 10.3917/vse.211.0070. URL : <https://doi.org/10.3917/vse.211.0070>.

- [62] Mehdi ZOUINAR. « L'IoT au service de l'interaction homme-machine ». In : *Interactions* 26.2 (2020), p. 32-41. URL : <https://www.revue-interactions.com/article/iot-interaction-homme-machine>.